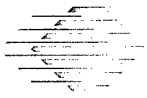
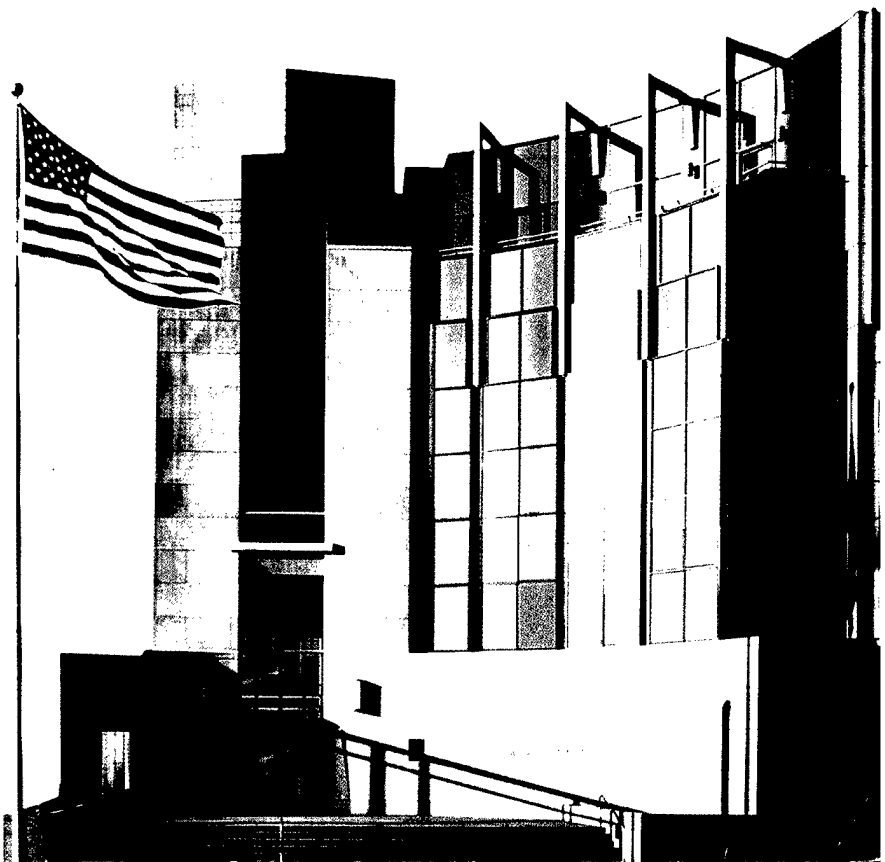


Carnegie Mellon
Software Engineering Institute

Third DoD Product Line Practice Workshop Report

Sholom Cohen
Brian Gallagher
Matthew Fisher
Lawrence Jones
Robert Krut
Linda Northrop
William O'Brien
Dennis Smith
Albert Soule

July 2000



TECHNICAL REPORT
CMU/SEI-2000-TR-024
ESC-TR-2000-024

Carnegie Mellon University does not discriminate and Carnegie Mellon University is required not to discriminate in admission, employment, or administration of its programs or activities on the basis of race, color, national origin, sex or handicap in violation of Title VI of the Civil Rights Act of 1964, Title IX of the Educational Amendments of 1972 and Section 504 of the Rehabilitation Act of 1973 or other federal, state, or local laws or executive orders.

In addition, Carnegie Mellon University does not discriminate in admission, employment or administration of its programs on the basis of religion, creed, ancestry, belief, age, veteran status, sexual orientation or in violation of federal, state, or local laws or executive orders. However, in the judgment of the Carnegie Mellon Human Relations Commission, the Department of Defense policy of "Don't ask, don't tell, don't pursue" excludes openly gay, lesbian and bisexual students from receiving ROTC scholarships or serving in the military. Nevertheless, all ROTC classes at Carnegie Mellon University are available to all students.

Inquiries concerning application of these statements should be directed to the Provost, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-6684 or the Vice President for Enrollment, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-2056.

Obtain general information about Carnegie Mellon University by calling (412) 268-2000.

Carnegie Mellon
Software Engineering Institute
Pittsburgh, PA 15213-3890

Third DoD Product Line Practice Workshop Report

CMU/SEI-2000-TR-024
ESC-TR-2000-024

Sholom Cohen
Brian Gallagher
Matthew Fisher
Lawrence Jones
Robert Krut
Linda Northrop
William O'Brien
Dennis Smith
Albert Soule

July 2000

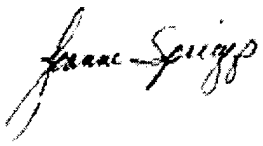
Product Line Systems Program

This report was prepared for the

SEI Joint Program Office
HQ ESC/DIB
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER



Joanne E. Spriggs
Contracting Office Representative

This work is sponsored by the U.S. Department of Defense. The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

This work was created in the performance of Federal Government Contract Number F19628-95-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 52.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Table of Contents

List of Figures

List of Tables

Abstract

The Software Engineering Institute (SEI) held the Third Department of Defense (DoD) Product Line Practice Workshop in March 2000. The workshop was a hands-on meeting to identify industry-wide best practices in software product lines; to share DoD product line practices, experience, and issues; and to discuss ways in which the current gap between commercial best practice and DoD practice can be bridged. This report synthesizes the workshop presentations and discussions.

1 Introduction

1.1 Why Product Line Practice?

Historically, software engineers have designed software systems for functionality and performance. A single-system mentality prevailed. Little attention was paid to the consequences of a design in the production of multiple software-intensive products or their long-term sustainment. Large software development, acquisition, and reengineering efforts undertaken with this single-system mentality perpetuate a pattern of large investment, long product cycles, system integration problems, and a lack of predictable quality. Each product involves vast investments in requirements analysis, architecture and design, documentation, prototyping, process and method definition, tools, training, implementation, and testing, with little carried forward to future products.

An increasing number of organizations are realizing that they can no longer afford to develop or to acquire multiple software products one product at a time. They have explicit needs to achieve large-scale productivity gains, improve time to market, maintain market presence, compensate for an inability to hire, and leverage existing resources. Many organizations are finding that the practice of building sets of related systems together can yield remarkable quantitative improvements in productivity, time to market, product quality, and customer satisfaction. They are adopting a product line approach.

A *product line* is defined to be a group of products sharing a common, managed set of features that satisfy the specific needs of a selected market or mission. It is most economical to build a software product line from a common set of assets.[1] In fact, the products in a software product line can best be leveraged when they share a common architecture that is used to structure components from which the products are built. This software architecture[2] capitalizes on commonalities in the implementation of the line of products, supports the needed variation among the products, and provides the structural robustness that makes the derivation of individual software products from software assets economically viable. The architecture and the components are central to the set of core assets used to construct and evolve the products in the product line.

---

1 A *software asset* is a description of a partial solution (such as a component or design document) or knowledge (such as a requirements database or test procedures) that engineers use to build or modify software products [Withey 96].

2 A *software architecture* of a computing system is the structure or structures of the system that consist of software components, the externally visible properties of those components, and the relationships among them [Bass 98a].

By product line practice, we mean the systematic use of software assets to assemble, instantiate, generate, or modify the multiple products that constitute a product line. Product line practice involves strategic, large-grained reuse as a business enabler. Some organizations have already experienced considerable savings by using a product line approach for software system production. Other organizations are attracted to the idea but are in varying stages of integrating product line practices.

In January 1997, the Software Engineering Institute (SEI) launched a technical initiative, the Product Line Practice Initiative, to help facilitate and accelerate the transition to sound software engineering practices using a product line approach. The goal of this initiative is to provide organizations with an integrated business and technical approach to the multi-use of software assets so that these organizations can produce and maintain similar systems of predictable quality and at a lower cost. One of the strategies for reaching this goal involves direct interaction with and nurturing of the community interested in product line practice.

This transition strategy has been executed in part by a series of product line workshops organized by the SEI. Four of these workshops, in December 1996, November 1997, December 1998, and December 1999, brought together international groups of leading practitioners from industry to codify industry-wide best practices in product lines. The results of these workshops are documented in SEI reports [Bass 97, Bass 98b, Bass 99, Bass 00]. These reports identify product line best practices, collectively refining and synthesizing some of the best ideas presented, and also identify issues that still require solution. In March 1998, the SEI hosted its first Department of Defense (DoD) Product Line Workshop, *Product Lines: Bridging the Gap—Commercial Success to DoD Practice.* Product line practices, DoD barriers and mitigation strategies, as well as similarities and differences between DoD product line practice and commercial product line practices were discussed and documented [Bergey 98]. A Second DoD Product Line Workshop was held in March 1999. This workshop marked a turning point from the SEI perspective in that the DoD participants talked about how they were implementing or going to implement product lines, as opposed to the familiar lament from past DoD forums that it would be impossible to implement product lines within the DoD [Bergey 99]. At both DoD workshops, the SEI was encouraged to continue to hold other DoD workshop events and to continue to bring best commercial practices to the DoD through these forums.

The SEI continues to refine the collective workshop results through work with collaboration partners, participation in other workshops, and continued research. In addition, the SEI is producing a framework for product line practice. The SEI's Product Line Practice Framework is the first formal attempt to codify comprehensive information about successful product lines. The framework describes the foundational product line concepts and identifies the essential elements and practices that an organization should master before it can expect to field a product line of software or software-intensive systems successfully. The framework organizes product line practices into practice areas categorized according to software engineering, technical management, and organizational management. These categories do not

represent job titles, but rather disciplines. The framework is a living document that is evolving as experience with product line practice grows. Version 2 of the framework was made available on the SEI Web site in August 1999 [Clements 99].

1.2 About the Workshop

The SEI held the third in a series of two-day DoD Product Line Practice Workshops in March 2000 to achieve the following goals:

- identify industry-wide best practices in software product lines
- share DoD product line practices, experience, and issues
- discuss ways in which the current gap between commercial best practice and DoD practice can be bridged

The workshop participants were referred to Version 2 of the SEI's Product Line Practice Framework to provide a common focus to structure the workshop presentations and discussions. All participants in this workshop were from the DoD acquisition and contractor community. They were invited based upon our knowledge of their experience with and commitment to software product lines as either DoD system acquirer or DoD system contractors. Together we elucidated and discussed the issues that form the backbone of this report.

The workshop participants included

- Peter Beck, TACOM/ARDEC, Picatinney Arsenal
- Robert Chadbroune, Litton-TASC
- Sholom Cohen, Product Line Systems Program, Software Engineering Institute
- Stephen E. Cross, Director of Software Engineering Institute
- Mark Dehlin, West Virginia High Technology Consortium (WVHTC) Foundation
- Bryan S. Doerr, Boeing - St. Louis
- Major Shirley Dominick, ESC/DIJ
- Margherita P. Eastman, The Aerospace Corporation
- Jack Ferguson, Director of Software Intensive Systems, OUSD (S&T)
- Matthew Fisher, Product Line Systems Program, Software Engineering Institute
- Michael Gagliardi, Dependable Systems Upgrade Program, Software Engineering Institute
- Brian Gallagher, Product Line Systems Program, Software Engineering Institute

- Janet Gorski, Scitor Corporation
- Russell Graves, MITRE Corporation
- Colonel Mick Hanratty, Director OS-JTF OSD
- Lawrence Jones, Product Line Systems Program, Software Engineering Institute
- Donna Jordan, DD-21/Anteon
- Judy Kerner, The Aerospace Corporation
- Bob Krut, Product Line Systems Program, Software Engineering Institute
- Victor R. McMillen, Joint National Test Facility
- Robert S. Miller, Altair Aerospace Corporation
- Major Aaron Moore, Joint National Test Facility
- Linda Northrop, Director, Product Line Systems Program, Software Engineering Institute
- William O'Brian, Product Line Systems Program, Software Engineering Institute
- Major Christopher Shotts, Joint National Test Facility
- Kimberley Simpson, Jet Propulsion Laboratory
- Dennis Smith, Product Line Systems Program, Software Engineering Institute
- Albert Soule, Product Line Systems Program, Software Engineering Institute
- Captain T. Ladson Webb, Director of Aviation Program, U.S. Navy

To properly set the context, the workshop began with three presentations by SEI technical leaders of the product line work. They characterized the current state of product line practice by describing the industry's best product line practices, the current contents of the SEI Product Line Practice Framework, and preliminary guidelines on developing a product line business case. Representatives from three of the participating DoD organizations then made presentations related to product lines within the DoD.

Following the presentations, the participants divided into four working groups. Three of these groups were to explore DoD product line practices in software engineering, technical management, and organizational management. They were asked to provide general comments on their category, especially as it is represented in the framework, and then select from among the practice areas identified in the framework for their category, and describe how each selected practice area "gets done" within the DoD community. The remaining group explored the explicit practice area, defining and communicating a business case. Some groups used the framework format to structure and document their discussion. Others were more free form.

The workshop concluded with the working groups presenting their results to the entire group, followed by a verbal evaluation of the workshop.

1.3 About This Report

This document summarizes the presentations and discussions at the workshop. As such, the report is written primarily for those in the DoD who are already familiar with product line concepts, most especially those who are already working or initiating product line practices in their own organizations. Acquisition managers and technical software managers should also benefit from the information in this report.

The report is organized into five main sections that parallel the workshop format:

1. Introduction
2. State of Software Product Line Practice: Digest of SEI Overview Presentations
3. DoD Software Product Line Experiences: Digest of DoD Presentations
4. Software Product Line Practices: Working Group Reports
5. Summary

The section following this introduction, *State of Software Product Line Practice: Digest of SEI Overview Presentations*, summarizes the three SEI presentations that set the context for the workshop. The next section, *DoD Software Product Line Experiences: Digest of DoD Presentations*, summarizes the three DoD presentations. Section 4 is composed of the four working group reports on selected practices in software engineering, technical management, and organizational management, and on defining and communicating a business case. Each of the working group reports reflects the interests, experiences, and style of the individual group. The emphasis and completeness of the information varies by group and by practice. The practices discussed are important in their very selection. The summary in Section 5 recaps the major themes and suggests future directions. Additionally, there is a glossary of terms at the end of this report.

2 State of Software Product Line Practice: Digest of SEI Overview Presentations

2.1 Introduction

Three SEI technical leaders in the product line work gave presentations aimed at setting the context for the workshop. Linda Northrop, Director of the Product Line Systems Program, led the session with an overview talk that highlighted the primary themes for the workshop. She reviewed the definition of a product line, the state of commercial product line practice, the relevance of product lines to the DoD, the SEI Product Line Practice Initiative, and the SEI Product Line Practice Framework

She then distilled the results of the SEI's Fourth Product Line Practice Workshop held in December 1999. She uncovered the issues and solutions related to tool support for product lines shared by experts from seven commercial organizations with real-world experience in developing and fielding software product lines.

Larry Jones then presented the SEI work on acquisition strategies for product lines. Finally, Sholom Cohen and Dennis Smith presented preliminary SEI work on developing and communicating a product line business case.

2.2 Overview: SEI Product Line Practice Framework – Linda M. Northrop, SEI

2.2.1 What Is a Product Line?

A product line is a group of products sharing a common, managed set of features that satisfy specific needs of a selected market or mission. This definition has been around a long time in the manufacturing world. For example, a telecommunications company may offer a number of cellular phones that share a similar market strategy and an application domain, thus making up a product line. It is well understood in the world of manufacturing that when you have a product line you take economic advantage of the common features of the products in the product line when you are building those products. Common product designs and parts are used. Assembly lines and automated tool support are set up.

It is only within the last 10 to 15 years that organizations developing software have taken advantage of the commonality among similar systems and have taken a product line approach. The most economical way to approach software product lines is to build a common architecture that is shared by the products in the product line and that is used to structure the components from which the products are built.

The architecture and components are central to the set of core assets[3] used to construct and evolve the products in the product line. In other words, a software product line can best be leveraged by managing it as a product family (as it has traditionally been called in the field of computer science). A product family is a set of related systems built from a common set of assets. For example, if the product line of cellular phones is built from a common architecture and set of common components, it is managed as a product family. When we refer to a product line, we always mean a software product line built as a product family. This particular use of terminology is not nearly as important to us as the underlying concepts involved, namely, the use of a common asset base in the production of a set of related products.

Product line practice is therefore the systematic use of software assets to modify, assemble, instantiate, or generate the multiple products that constitute a product line. Product line practice involves strategic, large-grained reuse as a business enabler. The key concepts are

- **the use of a common asset base**, with the architecture being the pivotal asset,
- **in the production**, according to a predefined and documented production plan,
- **of a set of related products,** whose scope has been clearly defined and validated with a business case.

2.2.2 The State of Product Line Practice

A number of organizations have achieved their product line goals. They have already gained order-of-magnitude improvements in efficiency, productivity, and quality through the strategic software reuse afforded by a product line approach. However, even more important than significant cost savings, product line practice enables an organization to get its products to market or field at the right time. Time has emerged as a critical success factor in a number of highly competitive product lines, such as cellular phones, pagers, and printers. If a product reaches the marketplace several months after its competitor, it may have lost its window of opportunity and become a failure regardless of its features or cost.

The Swedish naval defense contractor, CelsiusTech, turned to a product line approach in the development of their on-board ship command and control systems in the mid 1980s [Brownsword 96]. Their efforts resulted in a product line they call Ship System 2000 that now spans 12 classes of ships, from surface vessels to submarines, and has fielded more than

[3] Some organizations refer to the core asset base that is reused on systems in a product line as a *platform.*

50 ship systems from the same architecture and set of components. Among many other benefits that CelsiusTech has enjoyed with this product line is a reversal in the hardware-to-software cost ratio, from 35:65 to 60:20, that now favors the software.

A number of other companies have shown similar success using a product line approach. Hewlett Packard, which like CelsiusTech has been using a product line approach for the past 10 years, has collected substantial metrics showing 2-to 7-times cycle time improvements with product lines. On one project they were able to ship five times the number of products, that were four times as complex, had three times the number of features, and with four times the number of products shipped per person.

Motorola used a product line approach for FLEXworks, a family of one-way pagers. They have shown a four-times cycle time improvement with 80% reuse. Cummins Engine Co. uses a product line approach for the engine control software for their diesel family and cite an order of magnitude decrease in build and integration time since going to a product line approach. Among other commercial domains that have shown equally dramatic results are air traffic control (Thompson, CSF, Raytheon), commercial bank systems (ALLTEL), telecommunication systems (Ericsson, Nokia, Lucent, AT&T), college registration systems (Buzzeo), and consumer electronics (Philips). These organizations have not moved to product lines to break into the market. They have needed product line practice not only to improve time to market, but to continue their health in the market, to maintain market presence, to sustain unprecedented growth (especially poignant given today's employment market), and to compensate for an inability to hire.

Many more organizations are now attracted to the concept of software product lines to address their needs for faster, better, and cheaper software production. Before moving to a product line approach for software, an organization should first identify its business goals and then determine if product line practice is a viable strategy to reach those goals. Software product line practice is not a panacea, but it has demonstrated significant advantages in many organizations that had a business case to support a product line practice.

2.2.3 The Relevance of Product Lines to the DoD

There is a growing recognition within the DoD that new acquisition approaches leveraging best commercial practices need to be implemented. At the top DoD policy levels, acquisition reform from DoD Directive 5000.1 and DoD Regulation 5000.2-R have focused on using these best practices to reduce cost, schedule, and technical risks, and to advance architecture-based approaches to reuse that support open systems, interoperability, and commercial off-the-shelf (COTS) products. Statements by present and former top-level DoD officials all express a need for the DoD to leverage the best commercial practices that have turned around American commercial industry over the last decade. It is important for the DoD to use innovative, commercially proven practices to reduce cycle time, improve quality, reduce cost, improve efficiency, and reduce technical risks. At an operational level, it is not exactly clear how this will happen. Support is needed to understand what the commercially proven

practices are that cut cycle time and cost while improving quality and efficiency; what the viable architecture-based approaches to reuse are; and how systematic software reuse is adopted in a DoD organization.

There have been some DoD product line success stories that show results comparable to the commercial successes cited above. However, there are many other people within the DoD who are attracted to product line concepts but don't know how to proceed. Though progress has been made, we are not at the point where product lines are a truly viable, repeatable practice within the DoD: there is a gap between best commercial practice and routine DoD practices. Part of this gap is related to the standard acquisition approach of acquiring a single stovepipe system at a time, and part is attributable to the fact that the commercially successful practices have remained proprietary. The workshop summarized in this report is one of the planned activities of the SEI's Product Line Practice Initiative, which is attempting to bridge this gap.

2.2.4 The SEI Product Line Practice Initiative

The vision of the SEI Product Line Practice Initiative is that product line development will one day become a low-risk, high-return proposition and that techniques for finding and exploiting system commonalities and for controlling variability will be standard software engineering practice in the DoD, government, and industry. Our strategies to achieve these goals are to

- identify and mature product line practices of demonstrated effectiveness
- integrate and codify a business and technical approach to product line practice, accommodating multiple entry points, system types, organizational contests and domains
- provide materials for implementing product line practice
- build a community and an infrastructure to transition product line practice

One of the ways in which the SEI executes these strategies is to collaborate directly with organizations on product line efforts. The National Reconnaissance Office, the Joint National Test Facility, the U.S. Army Special Operations Aviation Technical Application Program Office, the F-22 Pilot Training Program, the Robert Bosch Corporation, and Caterpillar are some of the organizations that the SEI is working with to achieve success with software product lines. These efforts are aimed at maturing product line practices and targeted transition.

Widespread transition has been effected through a series of workshops and presentations, targeted at well-defined audiences. The SEI has hosted four workshops for commercial leaders [Bass 97, Bass 98b, Bass 99, Bass 00], organized seven workshops for researchers and technologists, and, with this workshop, hosted three for the DoD community [Bergey 98, Bergey 99]. Presentations have been given at countless government, commercial, and technical forums.

There are challenges, both technical and non-technical, that still need to be addressed:

- lack of widespread understanding in the DoD of software architecture and its connection to the business life-cycle and to other architectures
- no standard way to adequately represent architectures
- no codified architecture and product line migration strategies for the vast number of legacy systems
- few examples of acquisition strategies that support systematic reuse though product lines
- lack of repeatable, integrated technical and management product line practices

However, there are current trends that are highly supportive of software product lines—trends that reduce the risk of product lines and make a move to product lines more viable. These trends include

- the growing acceptance of the importance of software architecture
- the proliferation within major organizations of self-sustaining architecture centers
- the maturity of object technology
- the standardization of commercial middleware
- the growing popularity of the notion of "rapid development"
- community acceptance of well-defined processes for software development
- the growing acceptance in the software engineering community of the importance of product line practices and the rising recognition of the amazing cost savings that are possible

There is also considerable evidence of the growing maturity of product line practice. Universities have latched onto software product lines as an area of research. Software product line concepts are being taught in some universities. European inter-company collaborations have been established. Product line workshops are being organized, and the first product line conference is being held in August 2000. On the government front, the National Reconnaissance Office's Control Channel Toolkit (CCT) product line effort was completed on schedule, on budget, and with no outstanding risks or actions. CCT produced the following reusable assets:

- generalized requirements
- domain specifications
- a technical reference architecture
- component implementations

- test procedures
- a development environment definition
- a reuse guide

The first reuser of the CCT assets is achieving tremendous benefits in terms of lower costs, defect rates, and the staff and time required. These benefits are comparable to those of successful commercial product lines. CCT represents a government product line success story.

Government product lines are challenging in that the government is inherently an acquisition organization. However there are multiple options that a DoD organization can choose from in pursuing a product line approach:

- Scope the product line and develop the architecture.
- Acquire a product line architecture.
- Acquire the core asset base.
- Acquire a product built using product line technology.
- Acquire a product and some set of reusable assets.
- Acquire products built from a government asset base.
- Acquire an entire product line.
- Acquire products built from a non-government asset base.

Before a DoD organization selects one of these strategies, it should first be careful to understand the domain(s) involved in the product, to scope the product line properly, and to build a business case stating the goals to be achieved by the product line and the justification for the selected strategy. The DoD environment is currently much more positive about product line thinking, and several major DoD contractor organizations have begun product line initiatives.

The contexts for product lines vary widely in the nature of products, the nature of market or mission, organizational structure and culture, process maturity, technical skill, and existence of legacy artifacts. Nonetheless, the SEI has noted through direct customer collaboration, the workshops, and focused research, that there are some universal activities and practices that are key to successful product lines.

2.2.5 Product Line Practice Framework

We are capturing those essential activities and practices in the SEI Product Line Practice Framework. The framework is a Web-based, evolving document that is designed to address both development and acquisition contexts. The framework is primarily targeted at members

of organizations who are in a position to make or influence decisions regarding the adoption of product line practices.[4]

As depicted in Figure 1, core asset development and acquisition are distinguished from product development and acquisition, using these assets with the understanding that management orchestrates, tracks, and coordinates both sets of activities. The arrows signify the high degree of iteration involved.

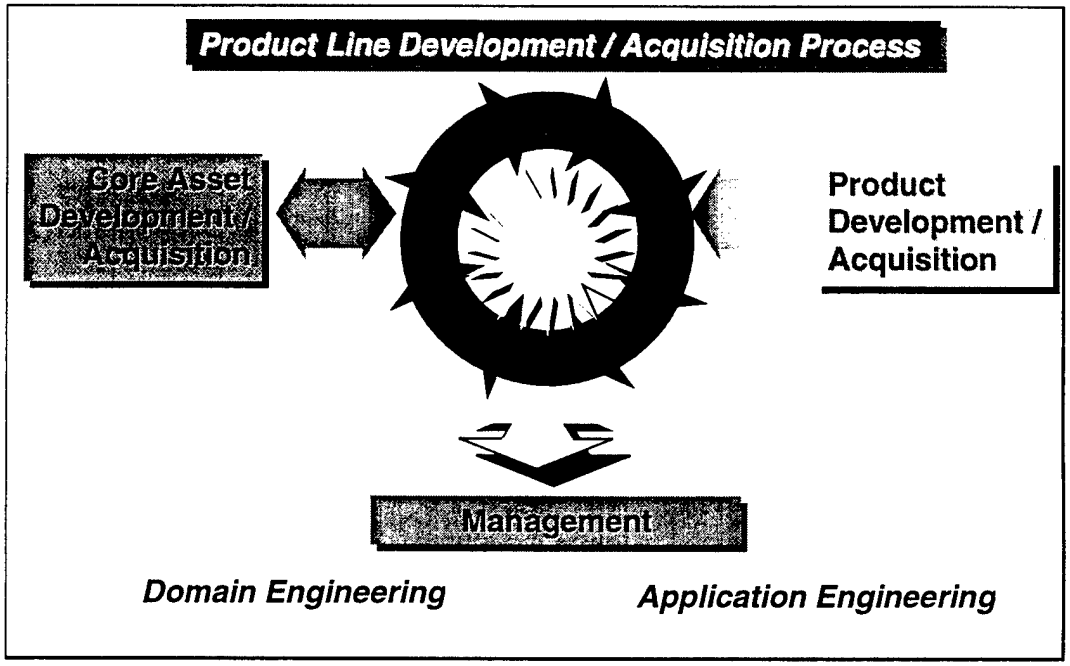


Figure 1: Essential Activities for Product Line Practice

On the left side of the figure, the critical core assets involved are the architecture and components. Inputs to the development and acquisition of core assets are product constraints found by analyzing the similarities and differences of current and projected products; production constraints such as might be found in a technical architecture; a production strategy for the assets; an inventory of pre-existing assets; and styles, patterns, and architectural frameworks. The outputs are the core assets, a preliminary list of the products they will support, and a production plan for how the core assets will be used in the development or acquisition of products.

On the right side of the figure, individual products are developed or acquired from the core assets using the production plan that has been established. Product requirements are developed and refined with the existing core assets in mind, and products that systematically reuse the core assets are output.

[4] At the time of this workshop, Version 2.0 of the framework was available on the SEI Web site. Workshop participants were asked to read Version 2.0. Version 3.0 is now available at http://www.sei.cmu.edu/plp/framework.html.

There is a strong feedback loop between the core assets and products. Core assets are refreshed as new products are developed. In addition, the value of the core assets is realized through the products that are developed from them. As a result, the core assets are made more generic by considering potential new products on the horizon. There is a constant need for strong and visionary management to invest the resources in the development of the core assets and to develop the cultural change to view new products through the filter of the core assets.

There are essential practices in a number of specific areas that are required to produce the core assets and products in a product line and to manage the process at multiple levels. The framework describes the essential practice areas for software engineering, technical management, and organizational management, where these categories represent disciplines rather than job titles. For individual practice areas, the framework provides

- an introductory description of the practice area
- aspects of this practice area that are peculiar to product lines
- how this practice area is applied to core asset development/acquisition
- how this practice area is applied to product development/acquisition
- specific practices in this practice area
- risks in this practice area
- references

2.2.5.1 Software Engineering

The software engineering practice areas include

- Domain Analysis[5]
- Architecture Definition
- Architecture Evaluation
- Mining Existing Assets
- Component Development
- Testing
- Requirements Elicitation, Analysis, and Tracking[6]
- COTS Utilization
- Software System Integration

[5] This practice area is called "Understanding Relevant Domain" in Version 3.0.
[6] This practice area is called "Requirements Engineering" in Version 3.0.

2.2.5.2 Technical Management

The technical management practice areas include

- Data Collection, Metrics, and Tracking
- Product Line Scoping
- Configuration Management
- Process Modeling[7]
- Planning and Tracking[8]
- Make, Buy, Mine, Outsource Analysis
- Technical Risk Management
- Tool Support

2.2.5.3 Organizational Management

Organizational management is the name we give to the management of the business issues that are visible at the enterprise level, as opposed to those at the project level. Enterprise management includes those practice areas necessary to position the enterprise to take fullest advantage of the product line capability. The organizational management practices include

- Achieving the Right Organizational Structure
- Building and Communicating a Business Case
- Funding
- Market Analysis
- Developing and Implementing an Acquisition Strategy
- Operations
- Training
- Customer Interface Management
- Technology Forecasting
- Launching and Institutionalizing a Product Line[9]
- Organizational Risk Management

---

[7] This practice are is called "Process Definition" in Version 3.0.

[8] This practice area is covered under two practice areas, "Technical Planning" and "Organizational Planning," in Version 3.0.

[9] At the time of this workshop, we considered this to be two separate practice areas: "Launching a Product Line" and "Product Line Institutionalization." Owing in part to workshop input and also to subsequent investigation and discussion, the two were combined in the published Version 2.0.

2.2.6 Future Direction of the Framework

The SEI Product Line Practice Framework is intended to be a living document. We made a decision to make it available before all of the practice areas are complete. Version 1.0 was the first step in engaging the community to provide feedback on the framework's accuracy and usefulness. We incorporated community feedback and our growing experience base in Version 2.0 and included a frequently asked questions section. We are encouraged to learn that more than 20 organizations have reported to us their use of the framework in their software product line efforts. The community response has been most favorable.

Future versions of the framework will build upon the current foundation, will continue to incorporate feedback and our experience, will complete the other practice area descriptions, and will describe a small number of product line scenarios.

To further support the transition of product line practice, the SEI is producing generic product line artifacts, case studies, technical reports, and instructional products, and is beginning to pilot product line technical probes in individual organizations to determine product line readiness.

2.2.7 Highlights of the Fourth Product Line Practice Workshop

In December 1999, the SEI conducted the fourth in its series of workshops on product lines. The goals of this workshop were to

- share information and issues about tool support for product lines
- stimulate the growth of a network of interest in software product lines
- populate the SEI framework with proven practices in the area of tool support
- identify gaps where experience is not properly reflected in the framework

Representatives from the following organizations were present:

- Robert Bosch
- Microelectronics and Computer Technology Consortium
- Thomson-CSFLCAT
- Electronic Data Systems (EDS)
- BigLever Software, Inc.
- General Motors Corporation, Powertrain
- General Motors Corporation
- Siemens
- Raytheon Systems Company

- Lucent Technologies

This workshop had a different character than our earlier workshops for commercial leaders. In the earlier workshops, we focused on product line practices in general. In this workshop we narrowed our attention strictly to tool support for product lines.

The workshop featured presentations by representatives from each of the participating organizations, followed by working groups to explore the practices and issues surrounding tool support for product lines from three, albeit somewhat overlapping, angles:

- What is special about tool support for product lines?
- How can tools be used to support product line efforts?
- What variety of tools is needed for product lines?

The working groups then presented their results to the entire group. The workshop concluded with a discussion of how to inform tool vendors about product line needs and how to motivate vendors to satisfy those needs. The results of this workshop were incorporated into Version 3.0 of the SEI's Product Line Practice Framework.

There were common themes woven throughout the featured presentation. There is obviously no completely suitable tool or tools for engineering a software product line nor does there seem to be a proposal for one. Making existing tools interoperate is critical and hard. Tools are no substitute for defined processes. Without robust underlying product line processes, tool support of any form is not helpful. Among the noted deficiencies in existing tools are the difficulty or inability to share information among simultaneous users, the lack of support for the variation points inherent in software product line assets, the inability of information provided by existing tools to support product line practice, and the difficulty in learning to use tools that are robust enough for use with product lines.

Tool support is needed for product line practices in the following traditional areas:

- requirements analysis
- architectural design
- component specification
- component internal design and coding
- product building
- verification
- configuration management and change control
- project management
- traceability

- documentation
- economic modeling
- application-specific commercial off-the-shelf infrastructure
- team communication and collaboration
- prototyping, rapid application development, modeling, and simulation
- measurement
- reengineering and reverse engineering

Tool support is also needed for

- environment building (tool-building tools and tool-integration tools)
- process modeling
- scoping a product line and capturing assumptions
- product deviation

The workshop participants concluded that there are differences in a product line environment that require not only automated support, but also robust and specific kinds of support. Because adequate tool support is currently not available, organizations modify existing tools, make their own tools, and do without. As a result, organizations take on risks that undermine their product line efforts and consume resources to work on mitigation strategies. There appears to be a real need to motivate tool vendors.

A more complete discussion of this workshop can be found in the corresponding workshop report [Bass 00].

2.3 Acquisition Strategies for Product Lines – Larry Jones, SEI

"Developing and Implementing an Acquisition Strategy" is a practice area in the Product Line Framework. This presentation followed the same structure used in the framework to describe each of its practice areas:

- Introductory Description
- Aspects Peculiar to Product Lines
- How Applied to Core Asset Development / Acquisition
- How Applied to Product Development / Acquisition
- Specific Practices
- Practice Risks
- References

2.3.1 Introductory Description

Acquisition may be applied to most, if not all, the other practice areas in the framework. This is particularly true in the DoD. Experience shows that virtually every organization and every acquisition is unique. Each organization has

- differing goals
- specialized missions
- particular assets
- unique requirements
- existing infrastructure
- prescribed policies and procedures

However, there is enough commonality in acquisitions to suggest some common product line acquisition practices.

The term "acquisition strategy" is used in many contexts. To set the stage for this practice area, it is necessary to define some key terms:

- *Acquisition* is the process of obtaining products and services via contract.
- A *contract* is a binding agreement between two or more parties that establishes the requirements for the products and services to be acquired.
- *Contracting* includes purchasing, buying, leasing, licensing, and procuring products and services.
- An *acquisition strategy* is a plan-of-action for achieving a specific goal or result through contracting for products and services.

In the DoD, an acquisition strategy may span one or more of the acquisition management life-cycle phases. In order for acquisitions to be successful for software product lines, the acquisition and program strategies must balance contractor interests and permit contractor participation in the product line efforts.

2.3.2 Aspects Peculiar to Product Lines

In the traditional DoD system acquisition process, the focus is primarily on acquiring an end product. There are often few constraints on how the design and implementation of this end product comes to fruition. Typically, this traditional process targets an "n systems - n acquisition" approach where there are "n" separate developments efforts and "n" separate maintenance efforts. Moving to a product line approach refocuses the acquisition to the strategic reuse of software common assets. This shift of focus constrains the acquisitions somewhat. The acquisitions and the acquisition strategy now must take into account product

line considerations, e.g., use of available core assets. These constraints must be reflected in the acquisition strategy. Strategic thinking becomes even more critical.

2.3.3 How Applied to Core Asset Development/Acquisition

For the product line core assets, acquisition involves commissioning suppliers or contractors to develop a software architecture; develop other core assets; mine legacy assets to extract core assets; manage, sustain, upgrade, and enhance the asset base *and* support product developers; purchase or license COTS components; or perform a combination of these activities.

2.3.4 How Applied to Product Development/Acquisition

For products made from the core assets, acquisition involves commissioning suppliers or contractors to

- develop a specific product *or* set of products from core assets
- maintain, upgrade, or enhance a product *or* set of products
- provide new assets (created during product development) for evaluation as candidate assets
- evaluate and incorporate core asset releases in products to ensure the integrity of the product line
- perform a combination of these activities

2.3.5 Specific Practices

In developing an acquisition strategy, there are questions that should be addressed by the acquisition manager.

When should we initiate acquisition strategy development?

> *Early!* Acquisition strategies for product lines should be developed as early as possible in the program or project to ensure sufficient coverage of all product line aspects of the initiative.

Who should be involved?

> *Team of key stakeholders!* These stakeholders include the end user, support personnel, and potentially the contractors or suppliers involved in the acquisition and the product line efforts.

How will the team gain an initial understanding of the acquisition role?

> The team can gain initial insight into the product lines and the role that acquisition will play through a draft product line operating concept (CONOPS). The CONOPS describes what the product line is all about, how it will be introduced, what

organizational elements are involved, how core assets will be obtained, and how products will be built.

What is involved in the planning, development, and implementation of an acquisition strategy for product lines?

In planning the acquisition strategy, the acquisition team must be aware of the levels of planning that occur for most DoD acquisitions, starting from the parent program, down through participating projects, to the product line organization, and finally to the acquisition team. Each of these levels imposes requirements to the lower levels that eventually influence how the product line acquisitions should and can occur.

As part of the planning activities, the acquisition team must develop the plan of action to support the types of and the needs of each acquisition contemplated. For example, the number and types of contracts should be established early. The planning should consider life-cycle continuity and possible multi-tier approaches; e.g., first acquire architecture, next acquire other core assets, then acquire products. The appropriate supporting language must be carefully included in the request for proposal (RFP) package.

While relating to most of the other practice areas in the framework, there is a particularly strong connection "Building and Communicating a Business Case," "Funding," "Market Analysis," and "Make, Buy, Mine, Outsource Analysis." All of these impact the acquisition strategy.

2.3.6 Practice Risks

Product line efforts have are inherently iterative. Many of the current DoD contracting practices make it difficult to support iteration. a product line effort will fail without support for iteration during development and sustainment. Creative approaches to acquisition that will accommodate the necessary iteration are needed.

Management visibility into the product line process is crucial for continued sponsorship. Without high-level sponsorship the product line acquisition strategy is likely to fail.

2.3.7 Summary

In summary, this practice area is critical to the DoD since a large part of the DoD mission is to *acquire* and field systems. This practice area interacts with many of the other framework practice areas. In fact, this practice area applies wherever an acquisition is contemplated for either the product line core assets or for the product development efforts.

Because of the uniqueness of each organization and the observation that every acquisition is unique, creativity on the part of the acquirer is required currently to successfully implement acquisitions as part of the product line approach.

2.4 Developing a Business Case for Product Lines – Sholom Cohen, SEI

A business case for product lines describes key organizational considerations that are necessary in making a go/no-go decision to pursue a product line approach. The business case is an important communications vehicle to identify product line goals and measures for tracking the move to the new approach. By documenting the expected costs, benefits, and risks of taking the product line approach, the business case supports determination of a course of action for management decision.

The next release of the SEI's Framework for Product Line Practice [Clements 99] will contain a practice area for developing a business case for product lines. The presentation outlined some current thoughts related to this practice area in an effort to set the context for the working group that would focus exclusively on developing and communicating a business case for a product line.

The following outline was proposed for the product line business case:

1. Background
2. Goals
3. Ground rules and assumptions
4. Desired situation vs. current situation
5. Identification of alternative business/acquisition strategies
6. Analysis of costs/benefits/risks of alternatives
7. Conclusions and recommendations

An example business case exercise was described. The example presented lessons learned with regard to having good historical data; reliability is essential to justify the business case. The example also addressed the need for relevance. The business case must address the goals and needs of the organization. If these have shifted during preparation of the business case, the results may not be useful or meaningful. The business case must speak to the correct audience. If presented incorrectly, the business case will have no impact. In either case, there will be no product line decision or there will be a no-go decision although facts may justify the business case as presented.

Once the business case has been defined, data must be collected and results measured for comparison against the goals stated in the business case and for future planning. Other ideas presented during the presentation are incorporated into the report from the business case working group found in Section 4.4 of this report.

3 DoD Software Product Line Experiences: Digest of DoD Presentations

3.1 Introduction

The following three presentations related to DoD software product lines provided the necessary DoD context that was helpful in framing the subsequent discussions:

- Scheduler Product Line: Architecture, Interfaces, Interoperability
- Bridging the Gap: Planning for Systematic Reuse in the Asset Sustainment Phase
- New DoD Acquisition Regulations and Product Lines

3.2 Scheduler Product Line: Architecture, Interfaces, Interoperability – Russ Graves, MITRE

In late 1998, the Joint Aerospace Applications Interoperability (JAAI) O-6 Steering Group first gathered to discuss interoperability issues and common development opportunities across the Command and Control (C2) Information Processing System (C2IPS), North American Aerospace Defense Command (NORAD)/ Unified Warfighting Support System (N/UWSS), and Theater Battle Management Core Systems (TBMCS). Representatives from the three Electronic Systems Center (ESC) system program offices and their respective user representative Major Commands (MAJCOMs) (AMC, AFSPC, ACC, AFSOC, AFRC, and AC2SIRC) participated in the steering group. Initial agreement among the group was to create a common application for unit-level scheduling, specifically to support scheduling activities for aircrew, aircraft, and space operations. The immediate focus was to reduce duplication of effort between two scheduling application developments, the Unit Level Planning and Scheduling (ULP&S) application and the Patriot Excalibur (PEX) Scheduler application, started by Air Force Air Mobility Command (AMC) and Air Force Special Operations Command/Air Force Reserve Command (AFSOC/AFRC) in early Fiscal Year 99. In addition, Air Combat Command (ACC) is acquiring the Squadron Aircrew Scheduling Application (SARA) as an interim solution addressing the short falls of the TBMCS scheduling application Theater Unit Level Scheduling Aide (TULSA).

After an initial evaluation of the scheduling domain, it was realized that there are unique variations in the scheduling activities, so a different approach than acquiring a common

application would be necessary. The C2 Centers and Applications (C2CA) Product Area Directorate (ESC/DIJ) proposed a software product line approach to develop unique scheduling applications that share common core scheduling capabilities and assets. The JAAI O-6 Steering Group tasked the C2CA Product Area Directorate (PAD) to develop a Scheduler Product Line by working with the C2IPS and TBMCS program offices.

3.2.1 Management Approach for the Scheduler Product Line

Since the C2CA PAD does not manage the current scheduling developments, does not have direct authority over the programs acquiring scheduling capabilities, and has limited funding to develop core assets, the greatest risk with creating an Air Force (AF) scheduling product line is organizational versus technical. To mitigate this risk, the C2CA PAD is attempting to establish management agreements with the ESC system program offices (SPOs) to ensure their commitment to a product line approach and to identify key activities in which they will participate. The C2CA PAD management approach is to leverage existing capabilities, developments, and research activities to define the basis for the product line's assets. The initial focus is on acquiring assets in three areas: architecture, interfaces, and interoperability.

3.2.2 Strategy for Achieving a Scheduler Product Line

Three areas provide an early opportunity to demonstrate the value of a product line approach: a common architecture, which is critical for achieving strategic reuse by specifying how independently developed components integrate; interfaces to external systems; and interoperability between scheduling applications.

To acquire a product line architecture, the C2CA PAD ideally wanted to get the contractors from the two scheduling application developments to work together on three tasks. Unfortunately, the TBMCS SPO had not officially selected the PEX scheduling application, so the C2CA PAD could not leverage their contact vehicle. They did however, align with the C2IPS SPO to get the ULP&S contractor to begin analyzing scheduling requirements and defining an architecture. The first of the three architecture tasks was to analyze requirements across the Air Force scheduling domains aircrew, aircraft, airspace, maintenance personnel, and space operations by partitioning the requirements into three categories: common to all, common to some, and unique. In the second task, the contractor would assess the aircrew and aircraft scheduling requirements specific to the mobility operations. Again, they would partition those requirements into the three categories. This partitioning would identify common requirements being developed in the ULP&S effort, which could be traced to potential core assets for asset mining. Lastly, the third task was to define an architecture that maximized the opportunities for reuse based on the identified common requirements.

For all of the aircrew and aircraft AF scheduling applications, there are common external interfaces required to obtain the necessary data for identifying the tasks and resource availability. For instance, all aircrew schedulers need to get the aircrew personnel data from the Air Force Operations Resource Management System (AFORMS) to validate their

currency requirements and track their flight hours. Currently, all aircrew scheduling applications define a specific interface with AFORMS to get the data for their type of personnel. By taking a product line approach, a single comprehensive interface is being developed such that all aircrew scheduling applications will interface with AFORMS in the same manner. Changes to AFORMS have an impact on only one common interface versus numerous unique interfaces. Another example is aircraft scheduling applications needing airspace availability information from the Military Airspace Management System (MAMS). To schedule an aircraft to fly through a certain airspace, the scheduling application needs to know the availability of the airspace. Again, a single comprehensive interface is being developed to reduce duplicative efforts.

Lastly, there is a need to improve interoperability between existing scheduling applications. The C2CA PAD has defined a common format for representing any schedule in Extensible Markup Language (XML) format. They are developing a proof-of-concept capability to demonstrate the use of this format mixed with information portal technology to distribute, manipulate, and display schedules to Web browsers. Based on the outcome of that effort, they can develop core assets to distribute, manipulate, and display schedules regardless of the schedule-producing application; thus, the assets become reusable across the product line.

3.2.3 Summary

Taking a product line approach to acquiring AF scheduling capabilities is very feasible. The C2CA PAD requirements and functionality assessment of existing scheduling applications indicates at least 75% commonality across aircrew and aircraft scheduling, which does not include non-functionality reusable assets such as architecture, test plans and procedures, configuration management, training, and user documentation. This approach can reduce the duplicative development efforts while still providing the flexibility to tailor the scheduling capability to the user's operational environment. As the common scheduling asset base increases, the ability to rapidly develop and deliver scheduling applications improves.

Several technical approaches have been debated. All have benefits and disadvantages, but all are equally viable solutions to achieving a product line. These technical challenges and issues can be addressed and hopefully resolved by developing a common architecture for the product line. Their ultimate success depends on getting the various scheduling application developers to agree on a product line architecture. The architecture is the most important asset for the product line.

Product line case studies show that funding and resources for core asset development and sustainment are typically controlled by a central organization. This is not the case with the Scheduler Product Line. Without authority over the current scheduling developments and no direct funding to acquire core assets, commitment to continue or participate in a product line approach can change at any moment. The up-front costs to build for strategic reuse are not secure and the justification for reuse is not guaranteed. Although the functional scope, common requirements, and technical approaches exist to build a solid business case for

taking a product line approach, the Scheduler Product Line effort is tremendously affected by a decentralized management approach.

3.3 Bridging the Gap: Planning for Systematic Reuse in the Asset Sustainment Phase - Robi Chadbourne, Litton-TASC

The Control Channel Toolkit (CCT) program successfully created a reusable asset base for spacecraft C2 systems.

This presentation focused on the transition to the asset sustainment phase. The presentation described insights gained from the Concept of Operations (CONOPS) for CCT. The insights are applicable to other DoD programs moving to a product line approach.

Asset sustainment is the critical phase where core assets are available for use in products, and these assets need to be maintained. The activities of this phase include

- use of the asset base in an operational setting
- correction of deficiency reports (DRs)
- maintenance of the architecture
- tracking of external developments
- addition of reusers
- improvement of the capabilities of assets

The primary purpose of the sustainment phase is to support the asset base, which consists of software components, the architecture, documentation, and processes. The synergy among these four elements facilitates the job of managing the assets by reducing the number and complexity of the interdependencies. An integrated management approach for these four elements is essential. If one element is not managed correctly, then long-term sustainment is likely to become more costly and error-prone over time.

3.3.1 Asset Sustainment Challenges

The transition from asset development to asset sustainment involves a fundamental shift in mind set from a single-user perspective to that of multiple reusers, or from one-time use to systematic reuse. The main challenges faced by organizations in this transition include

- obtaining adequate non-technical requirements for reuse
- addressing the critical technical issues of traceability of requirements and testing
- understanding other influencing factors

- understanding the importance of the architecture
- planning for change and early involvement with reusers
- availability of resources for systematic reuse
- senior management commitment

Frequently development efforts focus on technical requirements, but there must be some level of emphasis on the non-technical requirements for reuse. However, the CCT experience suggests that particular attention needs to be focused on non-technical issues, such as process definition and maturity, adequacy of testing, formation of architecture, metrics and reuser groups, definition of adequate metrics, and cognizance of development.

Critical technical issues such as traceability and testing were also discussed. The ability to trace requirements from mission needs to operational implementation is a critical technical need. It is important to establish traceability during the asset development phase. The practice of traceability has the benefit of enabling management and reusers to know how changes in the architecture or components affect requirements. It also helps management to enlist new reusers.

Comprehensiveness of testing is a part of both the development and sustainment phases. It has a direct effect on the confidence that potential reusers have in the ability of assets to meet their mission needs. Understanding the potential risks involved in less comprehensive testing is also of importance for reusers.

Other factors that can affect the asset base include industry and government standards, technology (hardware, software, communications, and connectivity), the environment in which the system operates, new candidate programs and systems, reuser needs (missions of current and potential reusers), and obstacles or threats.

A well-defined and well-understood architectural framework is a must for asset sustainment. An architectural framework constraining the number of interdependencies among components can simplify key asset sustainment functions such as problem determination, defect correction, regression testing, and component introduction/replacement/expiry. The lack of such a framework is not acceptable.

It is important to plan for longer term use of assets by multiple reusers and to plan for early implementation of the processes necessary to achieve reuse. However, adding reusers introduces the potential for conflicting needs, priorities, and schedules. There is a greater likelihood of success with a coordinated plan and schedule for changes and improvements. The early involvement of reusers is important to enable comparing their needs and requirements with the existing and planned asset base, as well as with requirements of other reusers. This will prevent the situation where subsequent reusers discover that their needs are not a good fit with the current or planned asset base.

In contrast, when a transition to the asset sustainment phase is made with only a single reuser, the program may not initially address the full scope of asset sustainment. In fact there is no systematic reuse if there are no prospects or intent to add other reusers. If one or more reusers are later added to the program, the transition from a single reuser to multiple reusers becomes a challenge. A multiple-reuser environment requires significantly different processes and planning than a single-reuser environment.

Investment in meeting these challenges is a must for long-term sustainment. Industry experience suggests that the lack of investment to sustain the total asset base diminishes the potential for successful systematic reuse.

3.3.2 Transition to Systematic Reuse

CCT transitioned to systematic reuse in the asset sustainment phase by

- looking at likely scenarios
- developing a set of questions to address the aspects of each scenario,
- developing a set of processes to answer these questions
- grouping scenarios and processes according to options
- incorporating processes into asset sustainment
- developing the CCT sustainment phase CONOPS from the scenarios, processes, and options

Several processes were developed by CCT with generic application to other programs. Some of these include multiple use, configuration management, sustainment management (component level), sustainment management (architecture level) and reuse management (metrics). The key challenges from the CCT experience included identifying the challenges, broadening understanding of the scope of asset sustainment and systematic reuse, developing the scenarios and processes, generating the asset sustainment CONOPS, and developing incremental options to handle varying resources for and commitment to systematic reuse.

3.3.3 Conclusions

A set of conclusions can be drawn from the CCT experience. It is important to recognize that the asset base includes far more than just the software components; it includes architecture, processes, and documentation. The total asset base needs to be addressed and managed for continued successful exploitation. Asset sustainment needs to begin in the development phase. Systematic reuse and the product line approach have a positive return on investment, but require up-front and continuing investment. Scenarios, processes, options, and the development of a CONOPS for the asset sustainment phase represent a solid starting point, but must be further developed and refined.

3.4 New DoD Acquisition Regulations and Product Lines - Jack Ferguson, Director, Software Intensive Systems, DUSD (S&T)

The primary function of the Deputy Under Secretary of Defense for Science and Technology [DUSD (S&T)] is one of oversight and assessment of the investment that the Department of Defense has made in science and technology (S&T) programs. The programs within DUSD (S&T) include High Performance Computing, DoD Modeling and Simulation, Laboratory Management/Security, Office of Technology Transfer, International Collaborations, Open Systems Joint Task Force, and Software Intensive Systems.

The DoD's investment in research and development programs was then discussed. The FY2000 investment in these programs is roughly $37.6 billion. Software development makes up nearly 42% of this investment. However, the amount spent on software assessment and oversight in the software S&T area is less than 1% compared to nearly 21% in other S&T areas.

This presentation focused on a draft re-write of the DoD 5000 regulation series—the policy and regulatory documents governing all DoD acquisition programs. The objectives of the update include the following:

- Develop a new acquisition model that reduces cost and cycle time while delivering improved performance.
- Move DoD closer to commercial-style approaches to acquiring systems.
- Implement Section 912 (part of the 1998 appropriations bill) recommendations.
- Implement other reports and key initiatives [e.g., Government Audit Organization (GAO) recommendations].
- Further streamline the acquisition process.

The proposed rewrites to the regulations are intended to allow a more flexible approach to acquiring systems. The language will explicitly encourage multiple process paths and emphasize that every acquisition is unique and can enter the acquisition process where it makes sense for the program. Policies will encourage acquiring systems using an evolutionary approach that includes reducing risk and developing new technologies prior to program commitment. Flexibility is allowed when constructing the phases of a program, and program managers and senior acquisition officials will phase funding commitment, program initiation, and requirements in-line with a true evolutionary approach. While the rewrite provides program managers more flexibility in developing systems, rigorous exit criteria will be enforced during the three milestone decision points: exploration, demonstration, and commitment.

Dr. Ferguson then described the proposed approach in more detail, outlining two major phases, a more flexible process for S&T and demonstration projects, and a more disciplined process for acquisition programs. The idea is to allow innovation in developing new techniques and advanced requirements and then to reduce the risk of incorporating these innovations through the use of Advanced Concept Technology Demonstrations (ACTDs) and Joint Warfighting Exercises (JWEs) to prototype how these ideas can be transitioned into acquisition programs. Once a decision to incorporate a technology into an acquisition program is made, a more disciplined process is followed to begin integrated engineering and production.

While the new concepts are promising and represent a major step toward implementing the objectives outlined above, there are still major issues to be resolved, and the details of policy rewrites may evolve as understanding of these issues matures. Some of the outstanding questions and issues include the following:

- Where does an acquisition program begin and what is its scope?
- When is full funding required? Should there be out-year wedges?
- How much integration is required during demonstration/risk reduction?
- How much information is needed to support milestone decisions?
- What is the role of S&T and the acquisition community in the transition to the acquisition phase?
- How does software fit in when acquiring systems using evolutionary concepts?

Dr. Ferguson next discussed where product line practices fit into the rewrite of the 5000 series regulations. DODI 5000.2 requires portfolio reviews. During these reviews, investments grouped by mission-related or business processes are evaluated together. The emphasis is on balancing sustainment, modernization, and research. These reviews will form the framework for managing a family of systems in a more systematic way allowing for the identification and maturation of a product line.

The presentation ended with a challenge for the DoD software community. Evolutionary acquisition of software-intensive systems requires creativity from practitioners to help harmonize software acquisition practices, milestone definitions, maturity models, metrics for product line management, process and architecture technology, and education and training. This challenge can't be solved at the DUSD level, but will require the entire community to work together and search for solutions that are actionable and innovative enough to allow for revolutionary improvements in how the DoD acquires software-intensive systems.

4 Software Product Line Practices: Working Group Reports

The following sections contain reports from the working groups. These working groups covered software engineering practices, technical management practices, organizational management practices, and developing a product line business case.

4.1 Software Engineering Practices

The software engineering working group discussed the relationship between system engineering and software engineering, the "Architecture Definition" practice area, and the "Mining Legacy Assets" practice area.

4.1.1 Software Engineering and Systems Engineering

The group began with a discussion of the relationship between software engineering and systems engineering. Two issues were initially identified:

1. the relationship between software engineering and system engineering
2. the relationship between software architecture and system architecture

These issues are not new issues, but they have become increasingly important as more and more traditionally hardware-only systems have become software intensive. The Framework for Software Product Line Practice focuses on software product lines and software engineering. It does not specifically address the relevance of systems engineering and system architecture. In systems with both software and hardware components, the system architecture and software architecture are closely interrelated and need to be considered together. The attributes of the software should be specified within the context of the system. Software tradeoffs should be made with a clear understanding of the system context. The software architecture should be validated against constraints of the system.

Often systems engineers do not understand software and do not appropriately involve software engineers in the system process. The group surfaced the following needs relative to the systems engineering/software engineering relationship:

- a better understanding of the business or mission requirements when developing a product

- a better understanding of the workflow, people, and skill set necessary to meet those requirements
- utilities for mapping software and hardware—specifically, utilities that simulate the software in a particular hardware configuration with the ability to be able to change the configuration and to determine how the software needs to change
- evaluation of different strategies for integrating system components
- focused discussion of the commonalities in the system
- visibility of the tradeoffs made between the hardware and the software
- traceability of the mapping between the hardware and the software—that is, an understanding of the hardware decisions that map to the software and vice versa
- an analysis of software at the same level as hardware in making systems-level decisions, especially in the context of a product line (For example, models of software with hardware characteristics could be used as input parameters to demonstrate the implications of specific hardware changes on the software.)

4.1.2 Architecture Definition

The "Architecture Definition" practice area was discussed in some detail. In defining the architecture, the group identified several issues to keep in mind:

- The architecture should be defined in the context of external systems, the organizational mission, and goals.
- When defining the architecture, the interrelationship between hardware and software to meet mission requirements should be described.
- Evolution of the architecture should be considered during the definition phase.
- The architecture definition should address placement of legacy components.

The group made a distinction between the solution space and the problem space, where the solution space describes the architectural decisions and the problem space describes the requirements levied on the architecture. In describing the architecture in the solution space, there is a need to have hooks that permit tailoring of the architecture. For example, the architecture may support different capabilities depending on the specific runtime environment. In the problem space, there is a need to allow for evolution and potential growth of customer requirements. The role of standards is in the solution space. The architecture should allow for absorption of changing standards.

Decisions to be made at the early stages of architecture definition include

- representation of components and interactions
- how the architecture fulfills mission requirements
- specification of quality attributes and tradeoffs

- demonstration of architecture feasibility
- plan for the use of metrics

When defining the architecture, there may be conflict between quality attributes. Balancing these attributes is a significant activity. In order to balance conflicting quality attributes, the most critical attributes must be determined early. This can be done by performing a breadth-first analysis to determine the most critical attributes early in the definition of the architecture, through examination of the essential business goals. Next, the attributes can be prioritized, and then a thorough analysis of the high-priority/critical attributes can be performed.

There should be a systems evaluation plan that ensures early validation of architecture flexibility in both the solution and problem space. Architecture evaluation should also address driving system quality goals. For example, if security is a primary system goal, an architecture evaluation should examine data aggregation since individual pieces of data that do not in themselves present security leaks could become security breaches when collected together. The issue of how to make the best use of existing legacy assets in defining the product line architecture remains (in the opinion of the group) unresolved.

The following non-functional issues also affect architecture definition:

- time requirements
- cost
- schedule
- use of COTS software

The group felt that products in the product line should be examined in terms of these issues as well as functional requirements, and considered it a risk if these non-functional objectives differed across the products.

The group strongly supported the framework's premise that the product line architecture is the most critical product line core asset.

4.1.3 Mining Existing Assets

The issue of legacy assets was brought up at several points during the group discussion, especially in relationship to the product line architecture. This section captures the group's views relative to legacy assets.

Decisions about product line software must be made in the context of existing legacy systems. A product line solution is often not affordable without the existence and use of legacy systems. The legacy system(s) provides some understanding of the domain—a necessary baseline from which to learn. Ideally, legacy components would be absorbed

initially into the product line and then migrated out over time. However, this may not be an easy task.

The product line architecture should absorb the legacy assets of coarse granularity, and thus granularity represents an important criterion for making decisions regarding legacy assets. Hidden costs in mining legacy assets for product lines can occur when legacy components need to be represented and used differently for the product line architecture than for their existing architectural context. For example, the legacy components may be decomposed functionally, whereas the product line architecture may require an object-oriented approach. This could result in a potential problem.

The existing legacy assets can serve an important function in documenting the current business because some organizations lose their corporate memory and the best source of information often resides in the legacy assets. However, it may be advantageous to rewrite the legacy assets if there is too large a mismatch with the product line architecture. In some cases, parts of the legacy components may not be required in the product line, especially if large components are being used.

Components that are volatile and mission-critical components need a more rigorous analysis. This is especially true if legacy assets are being used for these components. Prior certification of such legacy components is no guarantee that these components now work in the new context in which they are being used.

4.2 Technical Management Practices

The working group focusing on technical management practices began with an overview of the Product Line Practice Framework Version 2.0 and its technical management practice areas, which include "Data Collection, Metrics, and Tracking"; "Product Line Scoping"; "Configuration Management"; "Process Modeling"; "Planning"; "Make/Buy/Mine/Outsource Analysis"; "Technical Risk Management"; and "Tool Support." This introduction to the framework prompted much discussion on the contents of the technical management practice areas, including some challenges to the placement of practice areas. Once the group had a common understanding of the practice areas, the group used multi-voting techniques to decide which areas to discuss.

The multi-voting method resulted in the following two areas as the ones the group most wanted to discuss:

- Data Collection, Metrics, and Tracking
- Process Modeling

4.2.1 Data Collection, Metrics, and Tracking

The group decided that they wanted to answer the following questions:

- Are we just talking about instituting a metrics program?
- What are the goals (and business goals) of the metrics?
- What questions do I need to ask?
- How do I know if I achieve these goals?

Discussion of the above questions prompted an overview of the goals, questions, metrics (GQM) approach to measurement. The approach was elaborated and references were made to the SEI technical report on this subject [Park 96]. The group decided that this practice area as related to the product lines involved the following tasks:

- defining goals for management of the product line core assets or products
- determining questions to be answered, to know if the goals are achieved
- developing indicators to answer the questions
- collecting data to populate the indicators
- tracking or making decisions based on data

Some of the participants were concerned about the case where different products in the product line had different and conflicting goals. There was a recognition that in such a case there would need to be an effort at the organizational management level to resolve such differences for the good of the product line.

The group decided that the goals for the product line, for the development of core assets, and for the development of products were to be assumed as inputs into this practice area and that the output included a set of indicators, instantiating technical management decisions (replan, proceed, etc.).

Data collection should be started early in the product line effort, and the metrics collection should be kept simple. Metrics collection should be prioritized. It is better to collect a few good metrics than a large set that is too unwieldy or complex to be of use. The technical manager needs to build data collection into the development process and policies as early as possible, and then enforce the activity.

The group decided to clarify key terms for the purpose of their discussion and agreed upon the following definitions:

- **Data collection:** collecting the data for X (for your indicators/metrics)
- **Metrics**: information, representation of data

- **Indicators**: the inferences you gather from the metrics; the knowledge gained from the metrics, not the metrics or raw data
- **Tracking**: using metrics and indicators to help manage a project, meet goals, and support decision making

4.2.1.1 Aspects Peculiar to Product Lines

With a common understanding of the practice area and with common definitions, the group then shifted focus and decided to understand what aspects of "Data Collection, Metrics, and Tracking" were peculiar to a product line approach. The basic tasks associated with the practice area were deemed to be the same, but the real difference centered on how the goals may differ and the types of data you might need to collect.

Some goals of the core assets may be different than the goals required of the products. This difference might drive additional metrics to be collected. There may be unique data needs for sponsors/acquirers [i.e., return on investment (ROI)], product developers, and core asset developers. A product line needs additional sources of funding for maintaining the assets; therefore, the project goals may include organizational goals. Project-level goals and objectives may need to take into account the goals and objective of other groups. Groups involved in the product line may not have a consistent terminology set, so projects would need to standardize terminology to allow for consistent collection and reporting across projects.

One goal at the organizational level might be to enlist new users of the product line assets, requiring metrics geared at showing the value of strategic reuse versus developing unique functionality. Therefore, the product developers and core asset developers may need to collect specific data to encourage others to buy into the product line. These metrics would be centered around determining the cost to build and maintain the core assets and the cost to incorporate core assets into products. Potentially, data would need to be collected from multiple organizations, which have their own ways of collecting data, perhaps in different services or across organizational boundaries. The challenge would be to determine from the start how to standardize and use the data collected.

Core Asset Development Goals. Following the GQM approach, the group then brainstormed a list of potential goals that the technical manager of a core asset development team might have. The following list of goals was created to identify what data would be useful to collect:

- maximized reuse across products
- sufficient, robust documentation that supports the reuse and sustainment of the asset
- low number of defects found during reuse
- ease of integration
- robust (flexible) core assets

- reduction of development time for product developers
- reduction of training (development and end use) because of commonality
- balanced complexity (comprehensiveness versus reuse potential). You don't want the asset to cost more than the savings accrued by using it in product development.
- increased return on investment (ROI)

Each goal was then discussed, and a list of questions and/or indicators required were extracted that would help an organization determine if the goal was met.

Core Asset Goal 1: Maximized reuse across products

1. How much of the core asset base does each product use? How much of the asset base was used in a given product?
2. How much of the product is made from core assets?
3. Which core assets weren't used?

Core Asset Goal 2: Sufficient, robust documentation that supports the reuse and sustainment of the asset

1. How many discrepancies were found against documentation? How many change requests, modifications, and updates were required?
2. How many discrepancies were found because of incorrect use of documentation?
3. How long does it take to develop documentation?

Core Asset Goal 3: Low number of defects found during reuse

1. How many defects were found during reuse? What was the level of defect criticality?
2. How many defects were found during development? What was the level of defect criticality?
3. How many defects were found during product sustainment? What was the level of defect criticality?

Core Asset Goal 4: Ease of integration

1. What is the amount of time and resources necessary for asset integration?
2. What talent level is required to perform integration?
3. How much "glue" code must be built during product development to perform software integration?
4. What is the cost of asset integration during product development?

Core Asset Goal 5: Robust (flexible) core assets

1. How much core asset modification is required?
2. How much new code must be developed?

3. What is the amount of time and resources necessary for asset integration?
4. What talent level is required to perform integration?
5. How much "glue" code must be built during product development to perform software integration?

Core Asset Goal 6: Reduction of development time for product developers

1. How much time does it take to develop a product?
2. How much of the core asset base does each product use? How much of the asset base was used in a given product?
3. How much of the product is made from core assets?
4. Which core assets weren't used?
5. How many requests are made for additional variation points or additional capabilities?

Core Asset Goal 7: Reduction of training (development and end use) because of commonality

1. What is the cost of training for a specific product?
2. What is the cost of coming up to speed in multiple programs using the same common asset base?
3. How much time do individuals train across products in the product line?
4. How much training material can be reused across products?

Core Asset Goal 8: Balanced complexity (comprehensiveness versus reuse potential)

1. How much of a given product is made from core assets?
2. Which core assets weren't used?
3. What is the amount of time and resources necessary for asset integration?
4. What talent level is required to perform integration?
5. How much "glue" code must be built during product development to perform software integration?
6. What is the cost of integration?
7. How many requests are made for additional variation points or additional capabilities?
8. How many variation points are exercised in a given product?

Core Asset Goal 9: Increased return on investment (ROI)

1. How much does it cost to develop the core assets?
2. How much is saved in product development?
3. How much does it cost to integrate or reuse the core asset base?
4. What would it cost to develop a product from scratch?
5. How many products must be developed to justify the cost of developing both individual assets and the entire asset base?

Data must be gathered during both core asset development and product development in order to answer the above questions. The group cautioned that the list of questions they identified may actually be too detailed, and that it would be useful to prioritize the goals.

4.2.1.2 Product Development Goals

The group next focused on the goals of a technical manager in charge of product development. The goals would center on making project-level decisions. A technical manager of a product development team would need to be able to use quantitative data to decide whether or not to use the core assets. These types of metrics would support decisions typically used in make/buy/outsource/reuse decisions. A technical manager would also want to know the impact to his or her program (both positive and negative) from using the product line core assets to build products. More specific goals, followed by questions to answer and data to collect, are outlined below:

1. Product Goal 1: Update core assets, refresh rate.
   - What is the current state and evolution of the asset?
   - How long does it take to make changes to a core asset once a deficiency request is submitted?
   - What is the number of planned builds to core assets (major and minor)?

2. Product Goal 2: Minimize disruption of incorporation of core assets.
   - How much rework and re-integration effort is necessary to use the product line core assets?
   - How many defects in core assets are detected during product development?

3. Product Goal 3: Save time and cost.
   - What is the impact (cost, time, constraints) imposed on me to participate in the product line effort? For example, how much time must be spent feeding back assets developed as part of the product into the core asset base.

4.2.1.3 Risks

The group identified the following risks associated with the "Data Collection, Metrics, and Tracking" practice area:

- Too much effort is required to collect data, resulting in inappropriate use of resources or failure to perform the collection.
- Collected data are not used resulting in wasted effort.
- Data collected are not tied to decisions or goals, resulting in the inability to measure achievement of goals.
- Data are misunderstood, resulting in the inability to analyze metrics and track.
- Data are not timely, resulting in the inability to influence time-sensitive decisions.
- Collected data have limitations (e.g., trying to sell qualitative data as quantitative data, thus obfuscating results and allowing managers to make decisions that are unfounded).

4.2.2 Process Modeling

The group spent the remainder of their time discussing the "Process Modeling" practice area. Since there was no work available on this area in the framework, the group had difficulty deciding what this area really included. After exploring the idea of process modeling, the group agreed on the following definitions:

- **Process**: a set of activities to accomplish a goal
- **Process modeling**: developing a model of a proposed or existing process, exercising the model using scenarios, and evaluating the outcome. It is expected that some modeling technique would be used to indicate expected inputs and outcome, criteria, constraints, and available artifacts.

The processes and process models are themselves core assets. Modeling of the interaction and coordination processes involved in the product line effort is vital to success.

4.2.2.1 Core Asset Development Processes

The following list of potential processes to be modeled for core asset development were discussed:

- how to develop a core asset
- how to maintain the core asset
- how to extend the core asset
- how to integrate the core asset into a system
- configuration management of the core asset
- how to handle conflicts identified by product developers
- how to deal with the variation points in individual core asset types (for example, variation points in the product line architecture, variation points in components, etc.)
- how to maintain multiple versions of core assets
- how to deliver core asset updates to product developers
- how to prioritize core asset updates according to external pressures such as market changes versus internal pressures from product developers

4.2.2.2 Product Development Processes

The following is a list of potential processes a product developer might want to model:

- how to use core assets
- how to incorporate updates to core assets in existing products
- how to develop product-specific components to augment the existing core assets to meet specific product needs
- how to communicate with and provide feedback to core asset developers or other product developers

4.2.2.3 Risks

The group identified the following risks associated with the "Process Modeling" practice area:

- an inadequate process model resulting in lack of precise process guidance
- failure to develop or define a process, resulting in potentially unpredictable and unrepeatable results
- too much focus on modeling the process, resulting in lack of focus on the product itself (There is a fear that you could go too far with process definition and lose sight of the development effort. There is a need to keep scenarios and models as light-weight as possible while maintaining sufficient detail to guarantee repeatable results.)
- failure to evolve or improve the process, resulting in irrelevant processes
- defining too many processes, resulting in "process paralysis" (There is a need to balance process focus with product focus.)
- lack of consistency among the processes, resulting in disjointed efforts
- not articulating the process so all stakeholders understand how business is conducted, resulting in lack of coordination
- failure to follow the defined processes, resulting in an unsuccessful product line effort

4.3 Organizational Management Practices

The working group that focused on organizational management practices concentrated on two primary areas:

- the relationships among the organizational management practices areas
- the "Funding" practice area

4.3.1 Relationships Among Organizational Management Practice Areas

During the discussions, the group noted that there is a strong interrelationship among several of the organizational management practice areas, in particular, "Operations," "Achieving the Right Organizational Structure," "Launching and Institutionalizing a Product Line," and "Funding." Because these practice areas focus on enterprise-wide issues, this interrelationship is not entirely unexpected. However, the group felt that the relationships could be clarified. In particular, the "Operations" practice area seems to sit at a strategic level and might serve to orchestrate the other practice areas. If this were the case, then "Operations" could be expanded. Currently, "Operations" primarily emphasizes the initial fielding of a product line. The expanded view could more explicitly address institutionalization, evolution, and sustainment concerns of product line efforts. Thus, the "Operations" practice area might subsume "Launching and Institutionalizing a Product Line," or at least, the relationship between these two practice areas warrants clarification.

Next, the group developed scenarios to demonstrate the practice area relationships and their sequencing if an organization were establishing a product line effort. Following is a composite of those scenarios.

1. Execute a product line concept exploration phase, which includes the following activities:
   - Obtain funding and other resources for this phase.
   - Perform a market analysis.
   - Build a business case.
   - Determine whether to proceed and provide resources for the next phase. (The next phases are given assuming the decision is to proceed.)

2. Build the product line concept of operations. This embodies product line structures, processes, and the overarching plan for the product line. In particular, this includes determining the following :
   - organizational structures and how they interrelate
   - ownership and control of the product line
   - acquisition strategy
   - funding strategy and infrastructure
   - customer management strategy
   - training needs
   - risk management strategy
   - strategy for launching the product line

3. Execute the product line concept of operations. This includes the following activities:
   - Evolve the concept of operations and keep it up to date.
   - Acquire products and services according to the acquisition strategy.
   - Execute sustainment funding strategy.
   - Execute training strategy (both initial and sustainment).
   - Execute customer management strategy.
   - Manage risk.

- Forecast technology.

4.3.2 "Funding" Practice Area

The group next discussed the "Funding" practice area, which is not included in Version 2.0 of the framework. The discussion followed the structure for each of the practice areas in the Framework; that is,

- Description of the Practice
- Aspects Peculiar to Product Lines
- How Applied to Core Asset Development/Acquisition
- How Applied to Product Development/Acquisition
- Specific Practices
- Risks

The thrust of the discussions emphasized how these areas might be addressed in a DoD environment. The following considerations were suggested in completing the "Funding" practice area description.

4.3.2.1 Description of the Practice

The key to funding in the DoD is to get the program institutionalized in the Program Objective Memorandum (POM). At its core, this action requires the development of budget estimates needed for the program. Budget estimation requires consideration of the total life cycle of the program, multi-year requirements, and the "color" of money needed. Existing organizational assets that can be applied to the program must be identified and considered. Additionally, it is important to identify potential funding sources. An iterative development of the product line concept of operations can assist with these various activities.

4.3.2.2 Aspects Peculiar to Product Lines

In a product line approach, there are multiple potential funding sources. Thus, it is important to identify possible programs to participate and ways they could participate in the product line effort. Once this is done, an initial funding strategy can be developed. This might include assessing a tax on participating programs or obtaining research and development funds as seed money. Similarly, a funding strategy must be developed to cover sustainment and evolutionary efforts for the product line.

Because different programs will have different needs and the timing of these needs is unlikely to align with all product line activities, there are several issues to address:

- How are competing priorities among products to be resolved?
- How do these priorities affect the timing of the required funding?

- If programs are taxed for participation, how do the individual program priorities and timing affect the "fairness" of the taxation?

These issues highlight the need for additional, careful, up-front planning.

4.3.2.3 How Applied to Core Asset Development/Acquisition

Control and ownership of core assets is key because there must be an arbiter among competing program requirements. This makes definition and control of requirements difficult. If the prioritization and "fairness" issues are used as a basis, this affects how core asset development, sustainment, and evolution is funded. The following issues were identified:

- Who pays for the upgrade of core assets?
- What is the impact of upgrades on existing products?
- How long are versions supported?
- What are the backward compatibility guarantees?

4.3.2.4 How Applied to Product Development/Acquisition

The key question here is, "How do programs obtain and apply the funding?" Accurate funding estimates will require accurate data and estimation models based on experience in the context of a particular product line. The Product Line Concept of Operations should define how this is to occur.

4.3.2.5 Specific Practices

4.3.2.5.1 Pilot Approach

One working group member participated in a successful reuse effort by obtaining a pool of funding through various partners. Based on a business case, a partnership was formed among government programs, industry partners, and senior leadership within the Office of the Secretary of Defense. While this group of early technology adopters was willing to participate, better estimation tools are needed to bring in early- and late-majority technology adopters. Some details of this experience may be found on the following Web site. <http://www.acq.osd.mil/osjtf>.

4.3.2.5.2 Sponsorship Reinforcement and the Business Case

It is a fact of life that senior sponsorship is necessary to initiate and sustain a product line effort. It is also a fact of life that senior sponsors may be reassigned fairly frequently. Thus, sponsorship must be continually nurtured and reinforced. An important tool to reinforce sponsorship is a strong business case. Once a business case is developed, it is important to keep it up to date. Another strategy to maintain support for a product line effort is to see that the level of sponsorship extends down through the organization. Thus, there should be a deliberate effort to build and reinforce sponsorship through middle and lower management.

4.3.2.5.3 Incentives

It is important that the program and program managers have an incentive to adopt a product line effort. The group felt that all too often current practice provides disincentives. For example, when savings are projected or achieved due to introduction of technology, the program manager is typically "punished" by having these savings taken away. A specific practice to alleviate this situation was brought up by one group member who had the fortunate experience of being in a program where he was allowed to share in the cost savings.

The group also discussed the use of "negative incentives." A strong leader who wanted to institute a program of strategic reuse might effect this by constraining available funds for perceived stovepipe efforts and forcing the issue. It is also the reality that shrinking budgets tend to have this effect, but many organizations don't seem to be able to act even in the face of this threat. Eventually, organizations like these may go out of business.

4.3.2.6 Risks

Based upon their experiences, the group identified the following risks for the "Funding" practice area:

- A plan for a product line is approved, but adequate resources are not provided, resulting in insufficient funds to proceed successfully.
- Short-term needs are perceived as having higher priority, and funds are withheld or removed leaving the product line effort under-funded or not funded at all.
- With multiple funding sources, there is the risk that some sources will renege on their commitments.

4.4 Product Line Business Case

To support development of the "Building and Communicating a Business Case" practice area, participants in this working group related their actual business case or product line experiences to the following issues:

- current plans within participant organizations
- business case model in terms of process and products
- need for alternative models to account for different starting points in the life cycle or in context of the specific product line
- practice area as applied to core asset or product development

The goal of the working group was to explore these issues in order to contribute to the "Building and Communicating a Business Case" practice area. There is a significant degree of variation among product lines and product line organizations, and group discussions were intended to bring out those variations for the specific experiences represented by the

participants. The group included members with a mix of organizational backgrounds, including a DoD laboratory, a major DoD acquisition organization, organizations with both DoD and commercial consulting experience, and a federally funded research and development center (FFRDC) director and former Defense Advanced Research Projects Agency (DARPA) program manager. Discussion of key business case issues helped the participants gain insight into their own efforts and understand the differences that exist between individual product line approaches.

4.4.1 Business Case and Product Line Experience

Each participant provided a brief description of current activities and experience related to business case development or product lines.

4.4.1.1 Business Case Experience

What are the main topics that any business case must address? The business case must provide management with the information necessary to determine the "best" proposals, assuming that not all proposals are accepted. Criteria are often stated in terms of return on investment: the cost of making the technology change versus the benefit of the change. However, there are cases where cost is only one of several factors. The group proposed the development of a range of factors that can be used to assess the value of proposals to the organization. Such factors might include

- reduction in personnel required for integration
- time-to-field
- enumeration of short-term (quarterly cycle) and long-term (two-year or more) benefits

The rationale put forward by the business case must not only show how organizational goals are met, but must also propose a means to measure performance against those goals. The business case for product lines can take the general product line goals and, for each, provide example metrics and indicators for assessing performance.

The business case for organizations that are not primarily development oriented, for example DoD labs, will establish additional goals. These organizations are concerned with technology and are not limited to development and fielding of products. Their contribution to product lines may be to establish and document areas of domain expertise and to look to future products that are purely conceptual. Can they set goals and define need when products are information based? How do you structure a product business around user-configured applications to query data in a networked world?

The primary mission of a lab may be to develop concepts for producing and distributing quality data on the battlefield. The product line may be for capabilities that make data usable and available, including data collection methods that conform to the data source definitions and format. This business case must address the tradeoffs a customer makes in deciding

between local and remote data availability. The product line goal will be to ensure battlefield dominance, but how would this goal be measured?

Product line data may be for artillery, weather, intelligence, imagery, etc. In this case, the product line systems make available quality data that will take different forms depending on the customer. In cases where the data consumers are known, traditional databases may hold information that can be made available through a server. Where the consumers, format, and use of the data cannot be anticipated, the lab may need to consider alternative means for publishing the data they create. For example, users may be running simulations that use the data via the High-Level Architecture (HLA) to merge live and simulated data in training, logistics, planning, and exercises. Agent-based systems can be used to link the data with the customer. The business case for asset development must cover the cost of developing domain models or other data representations and building HLA federates; for product development, the business case may cover fee for services or other cost recovery methods.

Another DoD goal is to increase the pool of small developers involved in DoD acquisition via COTS or other approaches without using them as subcontractors on larger procurements. From the DoD perspective, this goal maintains the industrial base, increases competition to lower costs, and leads to a team effort for procurement via integrated product teams (IPTs). However, this change will not occur automatically for the following reasons:

- The DoD can't change the contractor culture unless there is a corresponding change to government procurement and systems management culture.
- The DoD must induce industry to adopt a process that should lead government to a better product, a product which is not necessarily cheaper. Process improvement programs are steps in the right direction but are not yet there.

The industry perspective on a business case is quite different. In commercial industry settings, the business case is concerned with funding and lines of responsibility. Proposals are ranked in order of value or priority and must account for setting and meeting their goals. This ranking is sometimes called rack-stack-evaluate. Multiple proposals may come from research and development, engineering, or marketing. A budget process looks at funding a number of proposals based on generation of sales and pricing. Sales managers issue recommendations regarding product attributes and performance, and an assessment of whether the product can sell at the price point.

The commercial business case must embrace a strong market perspective. A key question is, "What is the customer saying?" The answer may be in terms of product features, price points, potential sales, or competition. The market analysis and feedback from the sales force are critical in answering this question. The commercial organization must develop a valid, financial analysis as criteria for decision making. Standard measures such as net present value and discount rates help set the break-even cash flow and positive ROI points. Product lines are a long-term strategy and would be difficult to sell where even 18 months is beyond

the normal business cycle. However, most commercial product organizations would argue that they have product lines and that the normal business cycle reflects the release cycle for upgrades or new versions of existing products. The ability to sell proposals in the business case diminishes as the time period for break-even increases, unless it can be shown that the promised benefit is strategic.

The measures of profitability have their parallel in the DoD where cost reduction and avoidance are essential. The DoD seeks ways to accomplish more with fewer resources, and the business case to a DoD organization must address these issues. The business case must be keyed to the audience. While the DoD does not compete in the commercial sense, competition may be characterized in terms of addressing international threats. The market analysis for DoD product lines will look at potential threats, although in today's world, intelligence estimates are generally not longer than five years. The "Building and Communicating a Business Case" practice area must therefore provide specific practices for establishing the audience, assessing their needs, and addressing other business case issues.

For the DoD, the audience will consist of several levels, including services, program executive officers (PEOs), or the entire DoD for strategic investments. Investments for cost avoidance or to achieve other goals must be near-term in affect, generally within the program objectives memorandum (POM) cycle. While planning for the Future Year Development Plan (FYDP) goes out to six years, PEOs take a much shorter view in achieving cost reduction. There is a two-year management maximum dictated by the POM, so the business case must show the precise direction of investments to go beyond that cycle. Spending now to reduce future expenditures will be considered and may even be funded by applying a tax to all programs under a PEO.

Spending money to avoid later costs may be good for users, but it may not support the development and business growth for contractors. There is an inherent conflict between the near-term goals of government and the long-term goals of contractors. The product line approach could de-motivate contractor organizations in an environment where downstream competition for support and maintenance is the payback for low initial contract pricing. Product lines require up-front investment and continued funding, making it difficult for corporate decision makers to justify the lower initial price, without guarantees of long-term maintenance agreements.

The business case to a DoD audience will be competing for funding with 25-30 other candidates. The audience will expect

- accurate math
- near term (two year maximum) results (Any longer implies too much risk, so overall program may be shown in increments.)
- goals that address weapon systems

- measures that address fiscal and angst reductions

The audience does not expect precise numbers, but they need assurance that the answers are in the right quadrant. If the case is well made, the government may set goals so that the only means to achieve them is with a major shift as recommended in the business case. Therefore, the business case must address the potential for increased risks if the proposals are put into effect.

4.4.1.2 Product Line Experience

While there are general principles that can be applied in developing a business case, the group presented specifics from their product line experience. The experience base of the working group participants included the following mission or market areas:

- Ada mission-critical systems (cruise missiles)
- Ground-based spacecraft command and control
- Scheduling system platform based on Ascent technology
  - used a COTS tool as an underlying asset to create a product line for fielding systems for commercial airlines
  - used Oracle-based reference architecture, visible interfaces, data definitions/dictionaries
- Commercial product line software and hardware involving large information systems houses (e.g., integrating into systems across organization). A typical problem was obtaining agreement on basic terms such as "cash on hand."
- Army artillery systems: data and other information resources to support operations, testing, training, and simulation
- Naval shipboard information systems. Domains involved include imagery, logistics, weather, communications, video teleconferencing (VTC) support, remote technical advisor (maintenance support), medical, and intelligence analysis.

The group discussed the implications of moving to a product line approach for developing or procuring software. Much of this discussion aligned with results of previous workshops, but several highlights are noteworthy:

- Product lines are useful in justifying to line managers the need for higher up-front costs to build better design and system repositories. They would otherwise use people to get systems out the door in the old model of "better/faster/cheaper."
- Improving the efficiency of the platform (such as the Scheduler discussed in Section 3.2) does not have a benefit for the long range view of Army systems. In this view, integrated, network-centric systems will be the norm. Data requirements will be known, but not their specific use or even their source Applications will be required "just-in-time" as needed. The capability provided by the product line must be the rapid generation of applications in response to changing user needs. These may be agent based such that the data can

"find" the user or the user find the data. Data must include usage information to customize applications using it.

- For DoD labs, the "product line" will be the packaging and provision of domain expertise. This will be in the form of data that can be distributed throughout the battlefield, to training exercises, or to simulations. This results in a "compression" of time on the battlefield to provide an edge to the information holders.
- Navy shipboard systems have evolved from the early systems that did little more than control and move information, to systems that collaborate in a ubiquitous computing environment. Their development has similarly evolved from single contractors to a distributed development process managed by an integrator. Adding a product line approach today encompasses interactions with a variety of contractors regarding "standards" and generic applications. In advancing the state of practice in combat and information systems, there was tremendous growth in software and computing power, but no accompanying standards that would address building blocks to support either real-time or non-real-time applications. Tooling, standards, and commonality efforts remain at the implementation level, if they're performed at all.
- The scope of systems in this product line (or product lines) covers real-time weapons (direct and energy weapons) and non-real time. As a first step to product lines, the developers should establish architectures for these classes of systems and determine where and how they differ. In addition, the business case for new product lines must consider the demands of high-precision video teleconferencing for bandwidth in crisis events. This is a major risk area, because the DoD is increasingly dependent on commercial assets to support communications. This may not be adequate in a crisis where there is a demand surge, so the product line must include support for an infrastructure that can address this risk.
- The longevity of the next generation class of carriers is another element in the development of new naval product lines. The typical ship in the class will last 50 years. Overall, the class will span a period of 100 years from the launching of the first carrier in the class until the last is retired. While the ship itself will last 50 years, the major weapon systems will change over 3 or 4 times, the computing hardware 10 times, and the software 20-30 times. This longevity places enormous demands on the maintainability of the product line architecture. Historically, however, there has been no "center of excellence" for architecture decision making. Decisions have been relegated to lower levels with significant difficulties in managing the integration.

4.4.1.3 Issues

These introductory discussions raised the following issues, which were discussed further within the context of guidelines for the "Building and Communicating a Business Case" practice area:

- What measures and indicators will be used to assess performance in meeting goals established by the business case? Our experience from the Capability Maturity Model® (CMM®) is that improvements in the corporate maturity level increase quality, but the payback may not be measurable. What is the appropriate measure of return on investment and what is a reasonable time to expect positive return?
- How does the organization account for the cost of moving to a product line approach? In addition to costs for developing assets, the business case must deal with the effects of adopting processes for product lines, including the cost of training, incentives, and tool development or procurement. The way in which we account for costs must also be considered because there is currently no way to trace people working on core asset development versus traceability to the development of a specific product.
- What is the contractor's incentive to lower costs when profits are reduced? The savings realized by the customer as a consequence of reuse cut into the contract price and into profits. The business case is based on performing the same work with fewer people, following significant investment, which goes against the profit motivation of most contractors.
- How do we account for ongoing efforts in the business case? The organization may already be creating assets from ongoing efforts and from legacy products. Technology organizations may be developing pure data assets as a product line asset.

4.4.2 "Building and Communicating a Business Case" Practice Area

The group discussion next centered on contents of the "Building and Communicating a Business Case" practice area. The group determined that product line goals and measures are an important part of a business case and discussed how they should be covered. The discussion also looked at challenges and risks in developing a business case. In addition, the contents and approach for delivering the business case was discussed. The discussion took the next generation carrier development and treated it as a scenario. The group examined issues that must be addressed when developing a business case for product line development of shipboard systems.

4.4.2.1 Goals and Measures

The framework, along with other product line and systematic reuse studies, lists a great many benefits and goals of product line development. The business case should resolve which of these goals, or others, are the primary drivers for making the business case. The organization developing the business case should then set ways to measure performance against these goals and indicators of success. Armed with the goals to be achieved, measures for tracking goals, and the time table for achieving goals, it is possible to make a reasonable business case. As stated above, the audience is not looking for precise answers at this point, but they need to see the direction that the product line proposals are taking.

® Capability Maturity Model and CMM are registered in the U.S. Patent and Trademarek Office.

Table 1 lists goals that were proposed by the group. Each organization will have its own primary goals, probably three or four at most, and a second set of less critical goals. Beyond goal setting, however, the organization must set ways to measure whether the product line effort is meeting these goals. Measures should be trackable, though only estimates or projections may be available. The intent of this list is only to indicate the types of measures/indicators an organization should look for; it is not exhaustive.

The availability of information and its utility led to a discussion of the scope for a potential product line. The business case may use the results of a scoping activity or may apply scoping to set some ground rules for the product line. The group discussed several areas of potential variation that should be covered in a business case. The ability to support a range of potential users within a single product line will make the product line approach more attractive. These variations include

- context of capabilities of products in product line (e.g., Crusader, Paladin, M1An, M2An,...) within any one system
- variations of use including operational, training, maintenance, test, and evaluation
- geographic variation
- domestic versus foreign sales

The Department of Defense may have its own set of measures such as war-fighter security, and it may be necessary to develop DoD equivalents for the commercial notions of time-to-market or profitability.

Product Line Goals	Typical Measures	How to Track/Indicators
Cost savings/cost avoidance	Number of people in integration	Compare historical performance with product line performance
Time-to-field	Schedule	Total effort expended
Information (also business) dominance	Difference between the aggregate of information available to each of two opposing military commanders[10] (market share)	Ability to respond to threats (competition), new technology. Denial of technology to adversaries
Capturing domain expertise	Timeliness of bringing power to market	Response to threat or anticipated threat
Address/avoid crisis	Force assessment (business projections)	Maintain market share, competitive position, dominance
Assimilate and exploit new technology	Time to upgrade and field	Maintain dominance
Availability of information (remotely, accurately, timely)	Timeliness	Delivered when needed, customer satisfaction
Achieve financial success	Positive ROI	Profit margins
Ability to upgrade over life of product	Timeliness	Requirements met, cost of integration
Maintain competition	Contractor participation	Business expansion, limited contractor exclusivity, contractor partnering

Table 1: Product Line Goals, Measures, and Indicators

4.4.2.2 Challenges and Risks

Success in adopting the business case requires consideration of challenges to adopting the proposals as well as risks. The group discussions provided the following list:

- The organization developing the business case may come from government, industry, or industry with government inputs. The business case must be specific to the organization's goals and mission, as with other business cases developed by the organization. But because product line approaches are new and cover a wide range of organizational, technical, and managerial issues, the business case must draw on cross-functional resources from across the organization. This will require careful planning and intergroup coordination to meet critical milestones, including budget time tables and personnel availability.
- The organization must consider a number of cost factors in developing the business case. While there may be a good basis for estimating the costs of software development, the intangible costs of changing the process from single system to product line orientation

[10] David M. Link. U.S. Army white paper on Information Dominance, November 1995.

will be difficult to measure. Such costs will include training, incentives, tool development/procurement, and reorganization.

- Where maintaining competition or putting corporate partnering in place, there may be multiple business cases in play at the same time. Clearly each organization will have its own set of drivers and these may be in conflict. The opening discussion mentioned one such conflict: government's desire to reduce long-term sustainment costs, where industry tends to view that period as the time to recover some costs.
- The business case may succeed only if the product line is accepted by vendors. For example, the Control Channel Toolkit mentioned its projected use by a mix of contractors. The business case may be adopted by the organization, but it may still require selling to suppliers and customers.
- The business case must address the correct audience. This audience must include those who can make the final go/no-go decision for proceeding on the proposals contained in the business case. If given to the wrong audience, there will be no decision. If the business case does not address the needs of the decision makers, the decision will be no.
- The organization must have good historical data on which to base decisions. The business case must be able to compare past, current, and projected costs; time-to-market; market share; and competitor information to make the case. While it may be possible to estimate prior results, these will tend to weaken the argument.
- The business case must contain not only a set of proposals for moving to a product line approach, but it must also include a brief set of action items for putting it into effect. Short of this, there will be a necessary planning step once the business case proposals are accepted.

4.4.2.3 Process and Business Case Contents

As an alternative to the contents proposed in the presentation on product line business case (as summarized in Section 2.4 of this report), the group considered the following structure used by a DARPA program:

1. What are you trying to do?
2. How is it done now?
3. What is the new idea or concept?
4. What is the potential benefit?
5. What is the cost/schedule?
6. How can I tell? (measures of effectiveness)

Using either outline, there is a set of steps that emerged from the group discussions. These steps are not in a specific order:

- Make sure you understand the audience. This step requires an understanding of their value system in terms of time-to-market or financial consideration for commercial organizations, and an understanding of the theme of military dominance for the DoD.

Also, the developers of the business case must determine up front whether the audience wants a range of choices from which to make a selection or a specific decision/policy package on which to base the go/no-go decision. If a decision to adopt the product line approach has already been made, they are more likely to want a set of alternatives from which to choose a specific approach.

- Encourage new ideas by providing benefits to risk-takers. (Better align the benefit to accrue to the organization that proposes, implements, takes the risk.)
- Schedule for the planning (POM or commercial) budget cycle.
- Estimate the size of the development effort.
- Identify sources of funding based on practice in the "Funding" practice area.
- Identify the competing interests:
  - up-front savings versus maintenance phase
  - promise of future business to recapture investments versus recompeting requirements

In developing costs for the product line approach, there are a number of process or other life-cycle factors. These will cover training, human resources, infrastructure, and tooling. Life-cycle costs must reflect current information and data, and they must establish some degree of validity. For those organizations performing a formal cost analysis, using net present value or other estimates, the costs should be stated as ranges with some indication of degree of probability. Less formal methods will necessarily result in less rigorous cost estimates. Most organizations, in practice, have not demanded precise answers, but they must see the appropriate direction. They also want to know the "total cost of ownership" across the entire life cycle of a product line (from inception, to initial products, sustainment, and retirement of the product line).

4.4.3 An Example Scenario

The group determined that working through a scenario would be an appropriate way to uncover specific business case issues. The next generation carrier offered a good opportunity as there are potential product lines within the carrier, as well as across ship classes such as the current Nimitz class and the new DD-21 destroyer class.

The scenario is as follows: The Navy desires to purchase new shipboard systems via product lines for the next generation of aircraft carriers. Each carrier has an expected life of 50 years, with the overall life of the fleet at 100 years. While the ship hull exists throughout its life, major components, such as radar, may be upgraded every 10-15 years. New computers (as many as 3000 computers on each carrier) will be brought on-board every 5 years and software will be completely upgraded every 2-3 years, so there is a need to manage computing resources as a complete product. There is also a goal of mapping capabilities across ship classes to share or avoid costs.

4.4.3.1 Goals for the Example

The following primary goals were established through the scenario discussion:

- **Create a business environment conducive to competition**. The Navy wants to eliminate its dependence on a single contractor for each major system. There may be a single integrating contractor, and even that role may be recompeted. There will no longer be lifetime support contracts for individual systems.
- **Encourage corporate partnering**. The Navy expects its contractors to team in delivering systems. In today's marketplace, no single contractor has or can recruit the personnel to do major systems on its own. The lead contractor will be dependent on other contractors, possibly the very ones who lost the competition, to get the job done. The business case must justify the costs of applying this concept in a product line approach.
- **Maintain technological leadership.** The competitive edge in sea dominance will come from information superiority. That requires having the appropriate information available when required and being able to receive or push that information even in a crisis, where extremely high data requirement demands must be met.

The business case development must come up with a means to measure performance against these goals. These are not the typical business goals, so the business case may have to develop its own measures of success. There will be more traditional business goals, however, including reductions in crew size and training requirements. But, for the less tangible goals, the Navy and industry cannot use the same measures.

4.4.3.2 Assumptions for the Example

There are several assumptions that have been made in planning for the fleet. One of the most critical is the need for enormous amounts of bandwidth, at least during surge situations. However, there is disagreement as to the cost. The business case must pin down whether bandwidth comes at a price or virtually for free. Whatever planning assumptions and alternatives are made, they should be explicit, managed, and controlled. Some of the alternative paths are listed below:

- Communications bandwidth. Surge requirements are rare, so plan for nominal loads and handle routine communications during off-peak hours.
- Data compression. This reduces bandwidth requirements but increases on-board processing. This may offer a reasonable tradeoff since processing throughput (as opposed to communications volume) is a locally controlled resource.
- Local area versus isolated information management. Information includes what is on the ship, within the battlegroup, or available remotely, possibly via satellite.
- Storage capacity versus bandwidth issue. The more information that is available locally, the lower the bandwidth requirements, but the greater the need for maintaining currency as stored data must be constantly synchronized. The carrier information center must provide to local ships when they are not in the proximity of the carrier. This requires

satellite communications instead of line-of-sight communications, another possible area of variability.

- Commercial to supplement. How can commercial assets support surge requirements? Is it safe to assume they will be there in a crisis situation? Will opposing forces be vying for the same commercial assets.
- Modeling and simulation for performance evaluation. A necessary investment is to improve our ability to model communications links and traffic flow to be able to make performance assessments. Without these tools, there are simply too many unknowns. The business case should include the development or use of these tools in making assessments.

4.4.3.3 Design Tradeoffs

The group touched on some design tradeoffs relative to product lines. The following design issues for the product line scenario were discussed:

- Design for evolution of communication architecture. Whatever decisions are made now must be reevaluated every few years due to changes in the technology. The communications architecture (COMA) must allow for ease of evaluation to respond to the ever-changing technology.
- Infrastructure support. Possible product lines will include the computing infrastructure. This will cut across numerous ship classes and can be made specific to Navy requirements on top of already existing standards such as Command, Control, Communications, Intelligence, Surveillance, and Reconnaissance (C4ISR), Joint Technical Architecture (JTA), Defense Information Infrastructure Common Operating Environment (DII-COE), etc. This infrastructure constitutes a horizontal product line.
- Specific application areas. These include such areas as mission planning, telemedicine, training, and VTC support. There may be a variety of systems on board where product lines will play a role and, looking across ship classes, there will clearly be product lines.

4.4.3.4 Management Estimates

In developing costs, the business case should start with the work breakdown structure. The organization can then apply Pareto analysis to uncover the major drivers. The business case should examine automation of manual tasks as an alternative. For example, automated versus manual logging may be a consideration. The DoD must realize that it is sacrificing some ownership rights and must be able to justify this against claims made and substantiated in the business case. The overall Navy technical strategy for the fleet should link to the business case to determine the percent of the overall mission area covered by the product line approach.

The business case under this scenario has a diverse audience including

- PEO and user community (next generation carrier)

- PEO alliance across platforms (including Nimitz and DD-21)
- Test platforms (ships built to assess new technology)
  - SMARTSHIP (test platform of technology to apply across classes)
  - SMARTCARRIER (for cost reductions, examples: multiple tasks invest in sensors, decision support in planning)

During development of the business case, the Navy will also want to initiate development of the acquisition strategy. This would include sets for familiarizing industry with Navy plans. Industry will want to know the product line ground rules and assumptions, such as the number of systems to be built, the types of contracts, and how competition will work. The announcement to industry should seek multiple players. The Navy will want to know

- all potential centers of activity, so that industry can develop their own business cases
- how to get immediate feedback via proposals
- how to identify specific deficiencies in the plan and obtain solution with costs (Industry may respond via its own business case.)
- how industry can anticipate Navy needs via independent research and development (IRAD) business development

Vendors will respond by offering a competitive solution. But the government will want to use this opportunity to broaden the industrial base by attracting new, smaller suppliers. The government can expect to see subcontracting arrangements or other contracting vehicles. In addition, the government must offer means by which industry collaborations can share an architecture to meet multi-ship needs.

4.4.3.5 Scenario System Development

Specific contractor approaches must address how to "parse" a system (that is, functional decomposition from an acquisition perspective). The business case must define how the contractors are expected to relate the needs of one ship to the needs of all ships in a class or in the fleet. The needs include

- threat (real or anticipated)
- countermeasures and solutions proposed
- exploration/proof of concept
- joint funding/risk

Meeting these needs will entail costs to the government to bring industry on board. In addition, there are more specific constraints that must be set for industry. These will include

- the scope of C4ISR
- obtaining consensus for industry response
- architecture assumptions as to bandwidth and other systems issues

- specific application requirements such as mission planning (This area depends on imagery, timeliness and accuracy of information, and connectivity with other DoD assets.)
- other constraints involving the overall goal of driving down crew requirements through better management of work loads (Telemedicine and logistics support are areas to be covered through advanced VTC capabilities.)

4.4.4 Summary

The working group participants offered their opinions of the effectiveness of the discussions. Here are some of the comments:

- "Changed my parametric thinking"
- Need to keep focus on "Whose business case are we talking about?"
- Get into the business case of the supplier. Answer the question "How is product line approach in the best financial interest of supplier? Do my organizational constraints make a product line approach too hard?"
- Product line thinking requires shifting from control to sharing.
- Issues for further discussion
  - How can the DoD support supplier product line vision based on business case?
  - How can DoD needs be met in this new way of doing business?

5 Summary

The SEI's Third DoD Product Line Practice Workshop explored the product line practices of organizations in the DoD community in light of best commercial practices in software product lines. The presentations and discussions validated the pivotal pieces of the SEI's Product Line Practice Framework and provided feedback and additional content for currently defined practice areas as well as those that are in the process of being defined. Challenges within the DoD community were discussed.

The working groups focused on specific practice areas within software engineering, technical management, and organizational management, as well as the particular practice area, "Building and Communicating a Business Case." As in the previous two such workshops, the empirical and anecdotal evidence that the workshop participants brought to the discussion significantly enhanced our current understanding of the practices and issues as they apply to the DoD. Traditional DoD acquisition strategies are not naturally conducive to software product lines, but product line practice is possible within the DoD. Participants in the workshop identified the following guidance:

- Carry out a system threat assessment report.
- Get buy-in at multiple levels.
- Determine waivers needed.
- Lobby the right people at highest levels.
- Build a prototype and demonstrate it.
- Sell the architecture.

When phasing out existing systems by introducing a product line, there may be strong organizational implications; therefore, it is critical to get early management support.

Within the DoD there needs to be increased awareness about DoD product line activities that may be relevant. It is critical for the DoD to think more strategically and to share information and outcomes between different areas. These outcomes could help to prevent duplication and redundant development.

In an effort to expand both the information base and the DoD community interested in software product lines, the SEI was encouraged by the participants to continue to hold similar workshops.

The results of this workshop are currently being incorporated into the framework, which will continue to be refined and revised as the technology matures and as we continue to receive feedback and to work with the growing community of software engineers championing a product line approach. If you have any comments on this report and/or are using a product line approach in the development or acquisition of software-intensive systems for the DoD and would like to participate in a future workshop, please send email to lmn@sei.cmu.edu.

References

[Bass 97] Bass, Len; Clements, Paul; Cohen, Sholom; Northrop, Linda; & Withey, James. *Product Line Practice Workshop Report* (CMU/SEI-97-TR-003, ADA327610). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1997. Available WWW: <URL: http://www.sei.cmu.edu/publications/documents/97.reports/97tr003/97tr003abstract.html>.

[Bass 98a] Bass, L.; Clements, P.; & Kazman, R. *Software Architecture in Practice*. Reading, MA: Addison-Wesley Longman, Inc., 1998.

[Bass 98b] Bass, Len; Chastek, Gary; Clements, Paul; Northrop, Linda; Smith, Dennis; & Withey, James. *Second Product Line Practice Workshop Report* (CMU/SEI-98-TR-015, ADA354691). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1998. Available WWW: <URL: http://www.sei.cmu.edu/publications/documents/98.reports/98tr015/98tr015abstract.html>.

[Bass 99] Bass, Len; Campbell, Grady; Clements, Paul; Northrop, Linda; & Smith, Dennis. *Third Product Line Practice Workshop Report* (CMU/SEI-99-TR-003, ADA361391). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1999. Available WWW: <URL: http://www.sei.cmu.edu/publications/documents/99.reports/99tr003/99tr003abstract.html>.

[Bass 00] Bass, Len; Clements, Paul; Donohoe, Patrick; McGregor, John; & Northrop, Linda. *Fourth Product Line Practice Workshop Report* (CMU/SEI-2000-TR-002, ADA375843). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000. Available WWW: <URL: http://www.sei.cmu.edu/publications/documents/00.reports/00tr002.html>.

[Bergey 98] Bergey, John; Clements, Paul; Cohen, Sholom; Donohoe, Patrick; Jones, Larry; Krut, Bob; Northrop, Linda; Tilley, Scott; Smith, Dennis; & Withey, James. *DoD Product Line Practice Workshop Report* (CMU/SEI-98-TR-007, ADA346252). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1998. Available WWW: <URL: http://www.sei.cmu.edu/publications/documents/98.reports/98tr007/98tr007abstract.html>.

[Bergey 99] Bergey, John; Campbell, Grady; Clements, Paul; Cohen, Sholom; Jones, Lawrence; Krut, Robert; Northrop, Linda; & Smith, Dennis. *Second DoD Product Line Practice Workshop Report* (CMU/SEI-99-TR-015, ADA375845). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1999. Available WWW <URL: http://www.sei.cmu.edu/publications/documents/99.reports/99tr015/99tr015abstract.html>.

[Brownsword 96] Brownsword, Lisa & Clements, Paul. *A Case Study in Successful Product Line Development* (CMU/SEI-96-TR-016, ADA315802). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996. Available WWW: <URL: http://www.sei.cmu.edu/publications/documents/96.reports/96.tr.016.html>.

[Clements 99] Clements, Paul & Northrop, Linda. *A Framework for Software Product Line Practice, Version 2.0*. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, July 1999. Available WWW: <URL: http://www.sei.cmu.edu/plp/framework.html>.

[Park 96] Park, Robert; Goethert, Wolfhart; & Florac, William. *Goal-Driven Software Measurement – a Guidebook* (CMU/SEI-96-HB-002, ADA293345). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996. Available WWW <URL: http://www.sei.cmu.edu/publications/documents/96.reports/96.hb.002.html>.

[Withey 96] Withey, James. *Investment Analysis of Software Assets for Product Lines* (CMU/SEI-96-TR-010, ADA315653). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1996. Available WWW <URL: http://www.sei.cmu.edu/publications/documents/96.reports/96.tr.010.html>.

Glossary

acquisition The process of obtaining products and services via contract

acquisition strategy A plan of action for achieving a specific goal or result through contracting for products and services

acquisition plan The artifact that is typically used to document the acquisition strategy

application engineering An engineering process that develops software products from partial solutions or knowledge embodied in software assets

core asset A software artifact that is used in the production of more than one product in a product line

A core asset may be an architecture, a software component, a process model, a plan, a document, or any other useful result of building a system.

domain An area of knowledge or activity characterized by a set of concepts and terminology understood by practitioners in that area

domain analysis The process for capturing and representing information about applications in a domain, specifically, common characteristics and reasons for variability

economies of scale The condition where fewer inputs (such as effort and time) are needed to produce greater quantities of a single output

economies of scope The condition where fewer inputs (such as effort and time) are needed to produce a greater variety of outputs

Greater business value is achieved by jointly producing different outputs. Producing each output independently fails to leverage commonalities that affect costs. Economies of scope occur when it is less costly to combine two or more products in one production system than to produce them separately.

platform	Core software asset base that is reused across systems in the product line
product family	A group of systems built from a common set of assets
product line	A group of products sharing a common, managed set of features that satisfy specific needs of a selected market or mission area
product line practice	A system of software production that uses software assets to modify, assemble, instantiate, or generate a line of software products
product line architecture	A description of the structural properties for building a group of related systems (i.e., product line), typically the components and their interrelationships The guidelines about the use of components must capture the means for handling variability that is either discovered in the domain analysis or is known to experts. (Also called a reference architecture)
product line system	A member of a product line
production system	A system of people, functions, and assets organized to produce, distribute, and improve a family of products. Two functions included in the system are domain engineering and application engineering.
software architecture	Structure or structures of the system, which consists of software components, the externally visible properties of those components, and the relationships among them [Bass 98a]
system architectures	Software architecture plus execution and development environments

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	1. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	July 2000	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Third DoD Product Line Practice Workshop Report	F19628-00-C-0003

6. AUTHOR(S)

Sholom Cohen, Brian Gallagher, Matthew Fisher, Lawrence Jones, Robert Krut, Linda Northrop, William O'Brien, Dennis Smith, Albert Soule

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2000-TR-024

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116	ESC-TR-2000-024

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

The Software Engineering Institute (SEI) held the Third Department of Defense (DoD) Product Line Practice Workshop in March 2000. The workshop was a hands-on meeting to identify industry-wide best practices in software product lines; to share DoD product line practices, experience, and issues; and to discuss ways in which the current gap between commercial best practice and DoD practice can be bridged. This report synthesizes the workshop presentations and discussions.

14. SUBJECT TERMS	15. NUMBER OF PAGES
commercial product line practice, DoD product line practice, Product Line Practice Framework, product line workshop, software asset, software architecture, software product lines	66

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL